MESSM: A FRAMEWORK FOR PROTEIN THREADING BY NEURAL NETWORKS AND SUPPORT VECTOR MACHINES

A thesis submitted to Middlesex University
in partial fulfilment of the requirements for the degree
of Doctor of Philosophy

Nan Jiang

School of Computing Science

Middlesex University

September 2005

ABSTRACT

Protein threading, which is also referred to as fold recognition, aligns a probe amino acid sequence onto a library of representative folds of known structure to identify a structural similarity. Following the threading technique of the structural profile approach, this research focused on developing and evaluating a new framework - **M**ixed **E**nvironment-**S**pecific **S**ubstitution **M**apping (MESSM) - for protein threading by artificial neural networks (ANNs) and support vector machines (SVMs). The MESSM presents a new process to develop an efficient tool for protein fold recognition. It achieved better efficiency while retained the effectiveness on protein prediction.

The MESSM has three key components, each of which is a step in the protein threading framework. First, building the fold profile library--given a protein structure with a residue level environmental description, Neural Networks are used to generate an environment-specific amino acid substitution (3D-1D) mapping. Second, mixed substitution mapping--a mixed environment-specific substitution mapping is developed by combing the structural-derived substitution score with sequence profile from well-developed amino acid substitution matrices. Third, confidence evaluation--a support vector machine is employed to measure the significance of the sequence-structure alignment. Four computational experiments are carried out to verify the performance of the MESSM. They are Fischer, ProSup, Lindahl and Wallner benchmarks. Tested on Fischer, Lindahl and Wallner benchmarks, MESSM achieved a comparable performance on fold recognition to those energy potential based threading

models. For Fischer benchmark, MESSM correctly recognise 56 out of 68 pairs, which has the same performance as that of COBLATH and SPARKS. The computational experiments show that MESSM is a fast program. It could make an alignment between probe sequence (150 amino acids) and a profile of 4775 template proteins in 30 seconds on a PC with 1G memory Pentium IV. Also, tested on ProSup benchmark, the MESSM achieved alignment accuracy of 59.7%, which is better than current models.

The research work was extended to develop a threading score following the threading technique of the contact potential approach. A TES (**T**hreading with **E**nvironment-specific **S**core) model is constructed by neural networks.

ACKNOWLEDGEMENTS

I would like to thank my supervisors, Dr. Wendy Wu and Dr. Ian Mitchell for their patience, guidance, support and constant encouragement during the course of this work. I really enjoyed the inspiring discussions with them and their invaluable suggestions. I am also grateful for all the constructive feedback they gave me upon the draft of this thesis.

I wish to express my deep gratitude to Prof. Arne Elofsson from Stockholm Bioinformatics Center, Dr. Liam McGuffin from UCL, Dr. Kuang Lin from NIMR, and Mr. Edward S.C. Shih from Academia Sinica (Taiwan) for their valuable discussion during this research.

I am grateful to Prof. James Alty and Dr. Christian Huyck for their help to revise the draft of this thesis.

I wish to thank all my friends, Huong Le Thanh, KehKok, Kunbin Hong, Yu Qian and Yan Zhang, for their friendship making the last three years enjoyable. Special thanks to Mr. Leonard Miraziz, the technologist of our school, for his assistance in computer technical support. Thanks go out to all others once doing me a favor in various ways.

My husband, Keran Zhang, is greatly appreciated for his loyal love, thoughtful consideration and unceasing encouragement. Last but not least, I am greatly indebted to my parents for their breeding and endless inspiration.

I am grateful to the School of Computing Science, Middlesex University, for the financially support during my study in United Kingdom.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS AND PROGRAMS

1D	One **D**imensional
3D	Three **D**imensional
3D-PSSM	Three **D**imensional **P**osition-**S**pecific **S**coring **M**atrix
ALL	**A**cute **L**ymphoblastic **L**eukemia
AML	**A**cute **M**yeloid **L**eukemia
ANN	**A**rtificial **N**eural **N**etwork
BPNN	**B**ack-**P**ropagation **N**eural **N**etwork
BLAST	**B**asic **L**ocal **A**lignment **S**earch **T**ool
BLOSUM	**BLO**cks **SU**bstitution **M**atrix
CATH	**C**lass, **A**rchitecture, **T**opology, **H**omologue
CASP	**C**ritical **A**ssessment of techniques for protein **S**tructure **P**rediction
DNA	**D**eoxyribo**N**ucleic **A**cid
DASEY	**D**irectional **A**tomic **S**olvation Energ**Y**
EMBL	**E**uropean **M**olecular **B**iology **L**aboratory
FLASH	**F**ast a**L**ignment **A**lgorithm for finding **S**tructural **H**omology of proteins
FSSP/ DALI	Database for fold classification and structural alignments
FUGUE	Sequence-structure homology recognition method
GenBank	Genetic sequence database
GenTHREADER	Fold recognition method for genomic sequences
HOMSTRAD	**HOM**ologous **STR**ucture **A**lignment **D**atabase

MESSM	Threading framework proposed in this research, **M**ixed **E**nvironment-**S**pecific **S**ubstitution **M**apping
NMR	**N**uclear **M**agnetic **R**esonance
PAM	**P**oint **A**ccepted **M**utation
PDB	**P**rotein **D**ata **B**ank
PHD	Protein secondary structure prediction server
PROSPECT	Threading method, **PRO**tein **S**tructure **P**rediction and **E**valuation **C**omputer **T**oolkit
PROSTAR	**PRO**tein **ST**ructure **AR**chive
PSI-BLAST	**P**osition **S**pecific **I**terative **BLAST**
PSIPRED	Protein secondary structure prediction based on PSSMs
PSSM	**P**osition-**S**pecific **S**coring **M**atrix
QP	**Q**uadratic **P**rogramming
RBF	**R**adial **B**asic **F**unction
RCSB	**R**esearch **C**ollaboratory for **S**tructural **B**ioinformatics
RNA	**R**ibo**N**uleic **A**cid
RMSD	**R**oot **M**ean **S**quared **D**eviation
SCOP	**S**tructural **C**lassification **O**f **P**rotein
SPARKS	**S**equence, secondary structure **P**rofile **A**nd **R**esidue-level **K**nowledge-based energy **S**core
SVM	**S**upport **V**ector **M**achine
SWISSPORT	Protein sequence database
TES	Threading score proposed in this research, **T**hreading with **E**nvironment-**S**pecific **S**core
TUNE	**T**hreading **U**sing **N**eural n**E**tworks

* Standard one and three letter codes are used for amino acids.

* Brookhaven Database codes are used for PDB entries.

CHAPTER 1 INTRODUCTION

1.1 Overview

Since the start of the whole genome sequencing projects in the 1990's (Fleischmann et al., 1995; Bult et al., 1996) and the recent completion of the human genome project (Jasny and Roberts, 2003; Collins et al., 2003), both the nucleotide sequence databases (e.g. GenBank, Benson et al., 2000; EMBL, Stoesser et al., 2001) and the protein sequences databanks (e.g. SWISSPROT, Bairoch and Apweiler, 1996; Bairoch and Apweiler, 2000) have been growing at an exponential rate. This deluge of information has necessitated theoretical, algorithmic and software advances in storing, retrieving, networking, processing, analyzing and visualizing biological information. As a result, information science has been applied to biology, which has generated a new research field called *Bioinformatics*.

1.1.1 Bioinformatics

Bioinformatics is a scientific discipline that uses a computational approach to understand and organize the information associated with biological macromolecules (Luscombe, et al., 2001). In the beginning of the genomics era, bioinformatics was mainly concerned with the creation and

maintenance of databases for storing biological information, such as nucleotide and protein sequences. More recently, emphasis has shifted towards the question of how to analyse large data sets in order to ultimately present a complete representation of the cell and the organism, and to predict the interaction networks in cellular processes (Kanehisa and Bork, 2003).

1.1.2 Protein structure prediction

Since the first protein structure was crystallized by Perutz and Kendrew (Nobel Prize in Chemistry, 1962), protein folding remains the most complex problem in bioinformatics. This is due to the complexity of the three-dimensional structure of a protein, and the fact that the protein structure is defined by many degrees of freedom (Sternberg, 1996). Protein structure prediction is one of the most important tasks in bioinformatics because the three-dimensional (3D) structure of a protein determines its biological function - for reviews, see (Thornton et al., 1999; Orengo et al., 2001). Based on the knowledge of the correlation between the protein sequences and known proteins, the structure, function as well as the evolutionary features of unknown protein sequences can be predicted by computational methods.

There are various relationships between proteins, from the case of almost identical sequences to apparently unrelated sequences sharing only a rough three-dimensional structure. This presents a challenge for protein structure prediction. One method, excellent at finding sequence similarity, might not perform very well in the case of only a structural relationship (or vice versa). Based on the similarity between the query sequence and the proteins of known structure, three possible theoretical approaches to predict protein structure for a given protein sequence of unknown structure are available (Lesk, 2002; Gibas and Jambeck, 2001). They are:

1) Comparative modelling method (Peitsch, 1996; Schwede et al., 2003): This is focused on predicting a 3D structure of protein from the known structures of one or more related proteins. It is used for sequences with a high homologue (30% or more sequence similarity) in the Protein Data Bank (PDB).

2) Threading method (Jones, 1999; Kim et al., 2003): If there is an absence of a significantly similar sequence (sequence similarity less than 30%) with known structure, the threading method is used to identify the proper fold pattern which the sequence might plausibly adopt.

3) *Ab initio* method (Park and Levitt, 1995; Bonneau et al., 2001): This is based on the 'thermodynamic hypothesis', which states that the native structure of a protein is the one for which the free energy achieves the global minimum. Without using the template of a known protein, the *Ab initio* method is used in the case when the fold of the query protein is significantly different from any known protein folds.

1.1.3 Threading

The threading method has been recognized as an effective protein structure prediction method since the threading program PROSPECT (Xu et al., 2001) performed the best in the CASP4 (Critical Assessment of techniques for protein Structure Prediction) competition. The threading method ``threads'' a query protein sequence into a set of known structural templates (constructed based on proteins with known structures) and finds the most suitable sequence-template fit. During this process, a scoring function is applied as an evaluation criterion to assess the compatibility of the sequence to the template structure. To date, numerous

threading programs with different scoring schemes have been proposed (e.g. Bowie et al., 1991; Jones et al., 1992; Sippl, 1995; Russell et al., 1996; Rice and Eisenberg, 1997; Jones, 1999; Thiele et al., 1999; Xu and Xu, 2000; Shi et al., 2001; Mallick et al, 2002; Kim et al., 2003). However, for these distantly related query and template proteins sharing the same fold, it remains a difficult task to develop an accurate scoring function to reflect the diverse biological constraints. The threading methods with atom level structure environmental descriptions have been proven to be more accurate than those with amino acid residue level descriptions (Lu and Skolnick, 2001), but they require a higher computational cost.

1.2 Motivations and Objectives

1.2.1 Problems and challenges

Although threading has been shown to be a powerful method for protein structure prediction, the success of it often relies on expert human interpretation of the results (Karplus et al., 2001). Due to the genome sequencing projects, the gap widens between the number of known sequences and the number of experimentally determined protein structures. To decrease the disparity between the amount of available protein sequences data and the number of solved protein structures, it is essential that threading methods are fully automated if they are intended to be used for genomes annotating.

Several automatic threading methods have been developed so far, such as, GenTHREADER (Jones, 1999), 3D-PSSM (Kelley et al., 2000), FUGUE (Shi et al., 2001) and PROSPECT (Kim et al., 2003). These threading programs perform well in either fold recognition or sequence to structure alignments. However, the overall performance of these models is rather

disappointing. For example, the alignment accuracy for GenTHREADER (Jones, 1999) is comparatively low; FUGUE (Shi et al., 2001) can only recognize 25% of homologous protein pairs with high confidence (99% specificity); PROSPECT (Xu et al., 2001) runs very slowly for long query sequences because of the large amount of computation involved in the model. To design a fast, reliable and automated threading framework is the focus of this research.

1.2.2 Approaches

The aim of this research is to build a new framework - **M**ixed **E**nvironment-**S**pecific **S**ubstitution **M**apping (MESSM) - for protein threading. The proposed framework is expect to achieve a better efficiency while retain the effectiveness on protein prediction. Figure 1.1 shows the components of the proposed framework, MESSM.

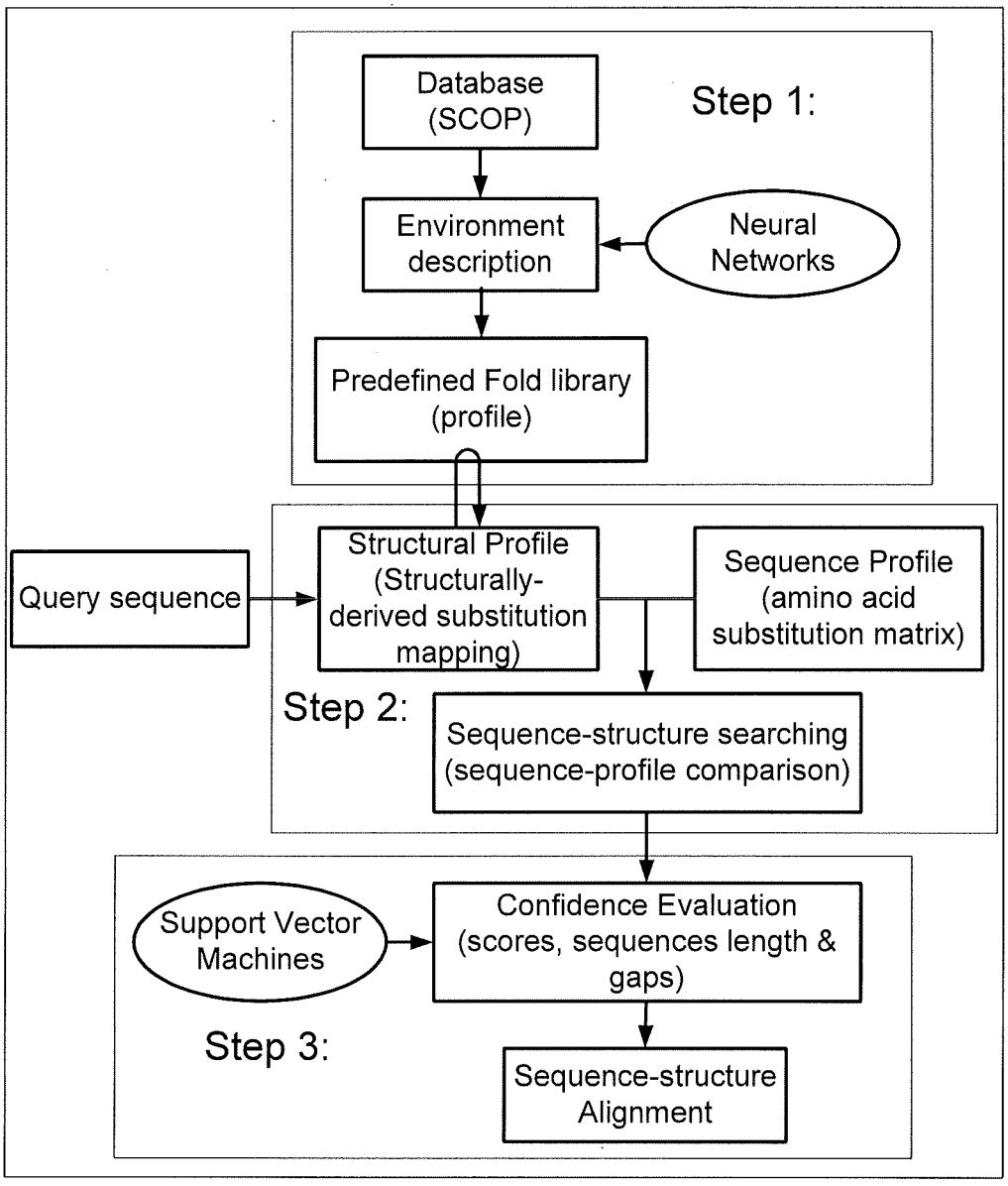


Figure 1.1 MESSM, a framework for protein threading. It includes three main steps. Step one is to build the fold profile library, step two is to linearly combine the structural profile with the sequence profile and step three is the confidence evaluation.

The MESSM has three main steps, briefly explained below:

1) *Building the fold profile library.* Given an amino acid residue with its environmental description, neural networks (NNs) are used to train the substitution probability of each pair of amino acids. A

predefined representative fold library is built as profiles on the substitution probabilities.

2) *Mixed substitution mapping.* According to consensus theory*, information is linearly combined from both the structurally-derived substitution score (obtained from the first part) and a sequence profile.

3) *Confidence evaluation.* A Support Vector Machine (SVM) is employed to measure the alignment significance between the protein query sequence and fold profile.

In summary, MESSM, uses machine learning approaches (ANNs and SVMs) in the framework to extract information from a large amount of data through a process of training from examples, and predictions on future test data.

1.2.3 Aims and Objectives

The focus of this research is on the cases where the protein query sequences do not have an apparent sequence similarity in the Protein Data Bank (PDB) thus comparative modelling methods for protein structure prediction cannot be used. There are a large number of proteins that belong to this case. According to Gerstein (1998), in a newly sequenced genome it is estimated that 30-50% of protein sequences can be detected to have weakly homologous with known protein structure.

This research proposes a new framework (MESSM) for protein threading

* The **consensus theory**, originated by Charles Sanders Peirce who called it pragmatism, and later pragmaticism, holds that a statement is true if it would be agreed to by all those who investigate it.

based on residue level environmental description. The use of machine learning approaches, ANN and SVM, enable more biological knowledge to be exploited in the prediction process compared with those statistically-based threading methods. The new framework for protein threading is expected to have a comparable performance to those more computational intensive, atom level structure environmental description models.

The proposed framework requires two properties:

1) Effectiveness: Using residue level environmental descriptions, NNs are adopted in the threading framework to extract more precise structural information of the protein. For such a framework, both the alignment accuracy and the fold recognition rate should be comparable to state-of-the-art structure prediction models. It should generate a higher prediction rate and better alignment accuracy than those models with the same residue-level environmental description. It is expected to have a comparable performance with those models using atom level structure environment description.

2) Efficiency: No heavy atom level pairwise contact potential is imported in the proposed threading framework so that a highly efficient dynamic programming algorithm can be used for alignment optimization. Also, as a SVM is used in the proposed threading framework for choosing the best template from the fold library, such a threading model should run quickly and automatically. Only a fast and automated protein structure prediction model is capable of matching the fast genome sequencing in the post-genome era.

The objectives of this research are:

- Design an effective residue contact measuring scheme based on protein residue level environmental description. The threading framework with atom level structure environment description needs more computational cost than the one with residue level description. To design a fast threading framework, residue level structure environment description will be used in this research. Furthermore, an effective residue contact measuring scheme need to be built to decrease computational load.

- Use NNs to train an amino acid substitution mapping. In contrast to those amino acid substitution tables designed by other researchers, a substitution mapping is given by NNs. By doing this, the prediction accuracy is expected to be increased.

- Build a representative protein fold library. Each representative fold in the protein fold library will be built as a 1D profile, which generated from the output of trained NN model. For each query sequence, the time used in finding the best match template will only depend on the time required by dynamic programming. Therefore, the MESSM can be a fast framework.

- Combine the protein structural profile with sequence profile. Sequence profile includes sequence evolutionary information. By combining structural information with sequences information, the MESSM is expected to be a reliable framework. That is, the fold recognition performance is expected to be retained if it can not be improved.

- Adopt a SVM to evaluate protein sequence-structure alignment. In contrast to the traditional expert human interpretation on

recognising the best fit templates, a SVM will be used to select the best template for each target sequence. By doing so, an automatic threading framework is expected.

- Validate the performance of the MESSM based on several benchmarks and compare the results with those of other current threading models.

1.3 Contribution to the Field

The main contribution of this research project is to propose a new framework for protein threading using a machine learning approach, and to outline the design and evaluation of the framework which uses a Mixed Environment-Specific Substitution Mapping as a scoring function. From the results, it is shown that the protein threading problem can be solved efficiently, in practice, by the MESSM. The detailed contributions of the research are summarized as follow:

1) **Residue contact measurement with residue level environmental description.** It has been generally agreed that the residue contact calculation is the most important factor in protein prediction models. Inaccurate calculation of protein residue contacts may reduce the efficiency of the model. This research proposes a new residue contact measurement with an amino acid residue level environmental description. It is built to reflect the fact that if the space between two amino acids is larger than one water molecule or a third residue, then they are too far to have contact. Thus, two kinds of contacts are considered, side-chain to side-chain contact and side-chain to main chain contact. The different measuring scheme of residue contact presented in this research is simple and

effective. The calculation of each residue contact is inexpensive. For each amino acid pair, only the distances between side chain centres and from α carbon to side chain centres are considered for computing.

2) **Environment-Specific Substitution Mapping generated by neural networks (Step 1 in Figure 1.1).** Given an amino acid with its structural environment, the NN is trained to predict the probabilities that it could be replaced by other amino acid types. In traditional amino acid substitution matrices or environment-specific amino acid substitution tables defined by other researchers, each structure position of a protein is defined as one of several groups according to the property of the amino acid. Unlike those approaches, each amino acid with its specific structural environment is described by its neighbour contacts and local structure in this research. This more precise structural information is extracted by the NN. Thus, the substitution probability of each pair of amino acids at any chosen structural environment can be generated and transformed into a log-odds score*.

3) **Representative fold library (Step 1 in Figure 1.1).** The fold library consists of 4775 representative folds - built on the basis of 3D-PSSM (Kelley et al., 2000). The size of the library is appropriate for experimental evaluation. A matrix of $n \times 20$ (1D profile) is built for each fold in the library to represent amino acid substitution generated from the trained NN model.

4) **Mixed substitution scores (Step 2 in Figure 1.1)**. According to the consensus theory, a mixed substitution score is proposed by

* The **log-odds score**, is the log-odds ratio, $s(i,j) = \log q_{ij}/e_{ij}$. Here "log" stands for natural logarithm. See formula 4.3 and formula 6.2 in this thesis for details.

combining the structurally-derived substitution score with the sequence profile from well-developed amino acid substitution matrices. The amino acid substitution matrices provided useful evolutionary information of protein sequences. The environment-specific amino acid substitution mapping generated by the NN is based on known protein structural information. Experiments show that the threading framework, with mixed substitution scores, has a better performance than the one with either structure or sequence profile only.

5) **The support vector machines (SVMs) approach for evaluation alignment accuracy (Step 3 in Figure 1.1).** After threading the query sequence to each template in the fold library, it is then required to choose the most probable templates for the structural model building. A SVM is employed instead of the traditional z-score (Flockner et al., 1995), p value (Karlin et al., 1990) or NNs (Jones, 1999) for the task of evaluation on alignment accuracy. The SVM approach is favoured for its effectiveness in choosing the correct templates over the other approaches.

6) **The TES (Threading with Environment-specific Score) score to measure the residue-structure compatibility**. Following the contact potential approach for protein threading (Jones et al., 1992; Bryant, 1996; Xu and Xu, 2000; Kim et al., 2003), this research is extended to design a threading score (TES) for measuring the residue-structure compatibility with the residue contact measurement used in MESSM model. A threading score is constructed by log-odds scores of predicted probabilities from a trained NN to determine which residue best fits its environment. Without the employment of contact energy commonly used in knowledge based potentials, the proposed threading score is demonstrated to be an effective score

on discrimination of native and decoy protein three-dimensional structure.

1.4 Organization of the Thesis

This thesis consists of seven chapters. Chapter one begins by introducing the motivation for carrying out this research. Then the objectives of this research are outlined. A briefly summary of the main contributions of this research is followed. The rest of the thesis is organized as follow.

Chapter two introduces some background knowledge concerning protein and machine learning methods. The three types of protein prediction methods are introduced. The concepts of two machine learning methods, NNs and SVMs, are presented. The applications of NNs and SVMs in bioinformatics are reviewed in this section.

Chapter three presents a detailed survey of the state-of-the-art protein threading models. A theoretical analysis of the protein threading is given. The research framework is proposed.

In Chapter four, the design of the MESSM framework for protein threading is discussed. A structural profile approach for protein threading is adopted. Three essential components of the MESSM are described. They are: the building of the fold profile library, the formulation of the mixed threading score and the SVM approach to perform fold recognition.

Chapter five evaluates the proposed MESSM with four benchmarks. They are: Fischer et al. (1996) test sets, ProSup benchmark (Domingues et al., 2000), Lindahl (Lindahl and Elofsson, 2000) data sets and Wallner et al. (2004) data sets. Both the alignment accuracy and the fold recognition rate

are tested and compared with the state-of-the art protein threading models.

In Chapter six, the research work is extended by using a contact potential approach for protein threading. A threading score is designed by NNs. The score function is tested by discriminating of protein native and decoys. The performance of the proposed threading score is evaluated by two benchmarks. The results are compared with the most recent and best performance threading scores based on energy potentials.

The final chapter summarizes the thesis, discusses the strengths, extensions and limitations of the research. Some suggestions are given for the directions of future work.

CHAPTER 2 CONCEPT AND THEORY

The concepts of proteins and protein prediction methods will be introduced in this chapter. Two machine learning methods, NNs and SVMs, are adopted to build the proposed protein threading framework. Therefore, the concepts of ANNs will be introduced with the emphasis on the BPNN model. A brief introduction to SVMs will be given in this chapter as well. An overview of the architecture of protein structures and the database for protein known structures is provided in Section 2.1. The summary of three categories of protein structure prediction methods together with the sequence-structure alignment algorithms are described in Section 2.2. Section 2.3 and section 2.4 contain an introduction to NNs and SVMs, as well as their applications in bioinformatics.

2.1 Proteins

Biochemically, all characteristic properties of life are affected by proteins (Lesk, 2002): for example, the conversion of chemical energy to mechanical energy, respiratory systems, photosynthesis, gene expression, genome replication, the immune system and the senses. Proteins participate in many different ways in these processes, and the precise tasks they carry out vary widely: they store and transport molecules (e.g. haemoglobin), catalyze chemical reactions (e.g. enzymes), transmit information between

cells, control the passage of molecules across cell and organelle membranes, bind to specific sequences of nucleic acids in DNA molecules, and they can simply act as structural building blocks.

Despite their diverse functions, all proteins are large molecules consisting of amino acids, the basic building blocks of proteins. The spatial conformation of a protein is dominated by the order of the amino acids contained in it, and their side chain chemical properties. Protein spatial conformations can be described at four different levels (Lesk, 2002):

1) Primary structure—a set of primary chemical bonds which build the amino acid sequence;

2) Secondary structure—the assignment of helices and sheets through the hydrogen-bonding pattern of the main chain;

3) Tertiary structure—the assembly and interactions of the helices and sheets;

4) Quaternary structure—the assembly of the monomer.

2.1.1 Protein Primary structure

Proteins are linear polymer chains of between tens to several thousands of subunits, where the subunits are 20 amino acids. All of these amino acids have a carboxyl group (COOH, also called C terminal), an amino group (NH_2, also called N-terminal), a central carbon (C_α) and a side chain (R) (Figure 2.1). The amino acids differ in the chemical composition of the side chain R, which contains between 1 (glycine) and 18 (arginine) atoms (see Table 2.1). Amino acids are connected together end-to-end in protein synthesis by the formation of peptide bonds between amino groups and carboxyl groups. Each amino acid in a protein is called the *amino acid*

residue or just *residue,* as its flanking atoms have been stripped off during the translation process.

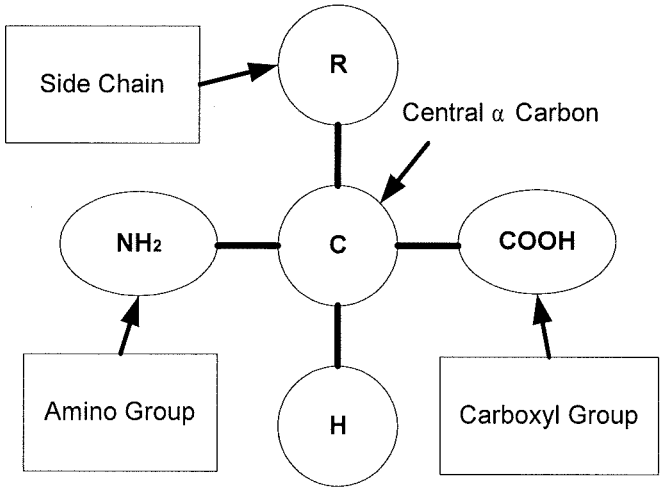


Figure 2.1Structure of an amino acid

Single letter code	Three letter code	Name	Residue mess (D)	Side chain
R	ARG	Arginine	156.2	$-CH_2CH_2CH_2NHCNH_2NH_2$
D	ASP	Aspartic Acid	115.1	$-CH_2COO$
E	GLU	Glutamic Acid	129.1	$-CH_2CH_2COO$
N	ASN	Asparagine	114.1	$-CH_2CONH_2$
K	LYS	Lysine	128.2	$-CH_2CH_2CH_2CH_2NH_3$
Q	GLN	Glutamine	128.1	$-CH_2CH_2CONH_2$
H	HIS	Histidine	137.1	$-CH_2$IMIDAZOLE
S	SER	Serine	87.1	$-CH_2OH$
T	THR	Threonine	101.1	$-CH(CH_3)OH$
Y	TYR	Tyrosine	163.2	$-CH_2$PHENOL
G	GLY	Glycine	57.0	-H
P	PRO	Proline	97.1	$-CH_2CH_2CH_2$ [N]
C	CYS	Cysteine	103.1	$-CH_2SH$
A	ALA	Alanine	71.1	$-CH_3$
W	TRP	Tryptophan	186.2	$-CH_2$INDOLE
M	MET	Methionine	131.2	$-CH_2CH_2SCH_3$
F	PHE	Phenylalanine	147.2	$-CH_2$PHENYL

V	VAL	Valine	99.1	-CH $(CH_3)_2$
I	ILE	Isoleucine	113.2	-CH (CH_3) CH_2CH_3
L	LEU	Leucine	113.2	-CH_2CH $(CH_3)_2$

Table 2.1 The 20 types of amino acids

It is generally assumed that a protein sequence folds to a native conformation or ensemble of conformations that is at, or near, the global free-energy minimum. All the necessary information for a protein to fold into its native secondary and tertiary structure is coded in its amino acid sequence (Anfinsen, 1973). Thus, it is fair to say that the three-dimensional structure of a protein is determined by its primary sequence. The problem of how the amino acid sequence determines the structure of a protein is called the protein folding problem (see Figure 2.2).

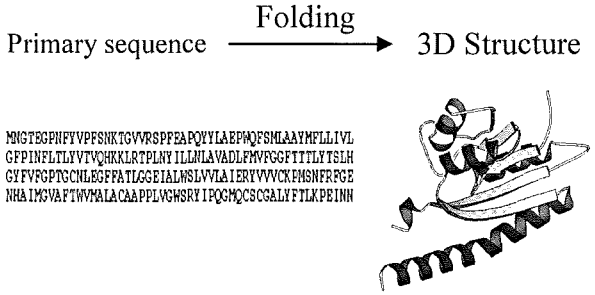


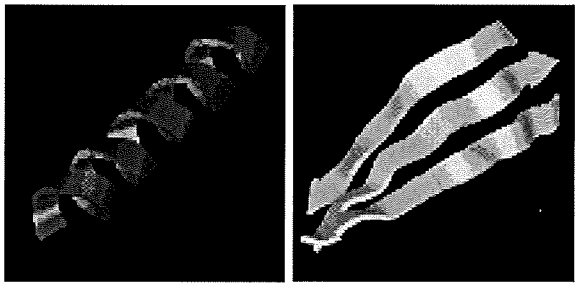
Figure 2.2 the protein folding problem

Proteins fold up into complex shapes due to the bonds formed between side chains. Not only are there strong bonds among two residues which are nearby along the primary sequence, but there can be strong bonds between two residues which are far away from each other. The former ones are called local interactions or short-range interactions; and the latter ones are called non-local interactions or long-range interactions. Interactions between two residues are also called pairwise contacts.

A segment of protein primary sequence can fold into a secondary structure because of the short-range interactions. Due to the long-range interactions, all secondary structures in a protein can form a specific tertiary structure with the loops connecting one secondary structure to another.

2.1.2 Protein tertiary structure

As mentioned above, proteins fold up because of the different properties of side chains. The different properties of side chains lead to five major inter-atomic forces that format the compact native tertiary structure of a protein. These are hydrophobic bonds, electrostatic bonds, hydrogen bonds, van der Waals force and sulphur bonds. A protein tertiary structure is hierarchically organized (Honig, 1999). The highest level is constituted by the complete protein, which can be considered through domains to secondary structures. Domains are stable, compact evolutionary units of a protein structure, which can fold autonomously and perform their functions semi-independently (Bork, 1991; Holm and Sander, 1998). Protein secondary structures are continuous fragments in a protein sequence showing distinct geometrical features (Ramachandran et al., 1974). Two basic secondary structures are the α helix and β strand (see Figure 2.3). Their structural features can be easily recognized (Kabsch and Sander, 1983). The major database that stores the 3-D coordinates of each atoms of protein is the Protein Data Bank (Bernstein et al., 1977; Berman, et al., 2000).



(a) Alpha helix (b) Beta strand

Figure 2.3 The basic secondary elements of a protein

Protein structure determination is the first step towards understanding of its function. Protein misfolding sometimes may cause fatal disease in organisms. One of the challenges for bioinformatics is to predict the protein structure and extract useful biological information, regarding its

biochemical function and role in the organism, from its amino acid sequence.

2.1.3 Protein Data Bank (PDB)

The Protein Data Bank (PDB) <http://www.rcsb.org/pdb/ >, originally established at Brookhaven National Laboratories in 1971, is now managed and maintained by the Research Collaboratory for Structural Bioinformatics (RCSB). At the start the archive only held seven structures of macromolecules and was only distributed by magnetic media. When the technologies of nuclear magnetic resonance (NMR) and crystallography for structure determination improved in the early eighties, the number of entries increased exponentially (see Figure 2.4). Now, the database contains all publicly available three-dimensional structures of proteins, nucleic acids, carbohydrates, and a variety of other complexes experimentally determined by X-ray crystallographers and NMR spectroscopists. An example of a PDB structure summary web page is shown in Figure 2.5. As of May 2005, the database holds about 30857 structures and is continually being updated.

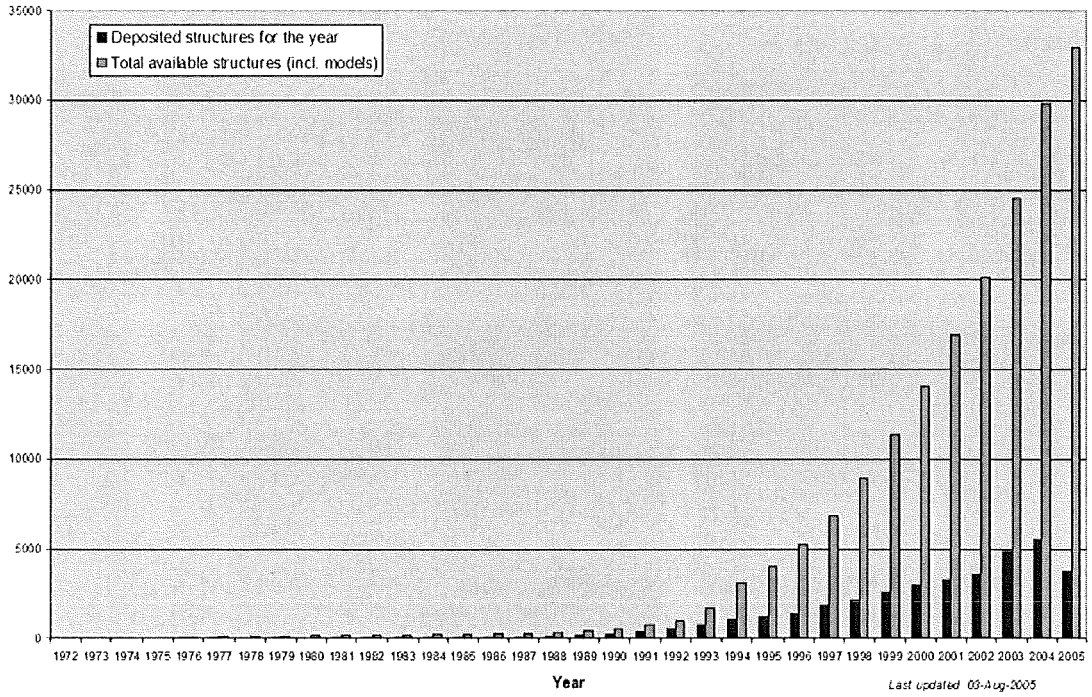


Figure 2.4 the growth of protein data in PDB (from http://www.rcsb.org/pdb/holdings.html)

Figure 2.5 An example of structure query from RCSB with the structure 1NRE (Nielsen et al., 1997).

2.1.4 Protein structure comparison and classification

Several publicly available classifications of protein architectures are available including SCOP (Structural Classification of Proteins; Murzin et al., 1995; Lo Conte, L., et al., 2002), CATH (Class Architecture Topology Homology; Orengo, et al., 1997) and FSSP/ DALI (Families of Structurally Similar Proteins; Holm and Sander, 1997). The aim of protein structure classification is to provide a detailed and comprehensive description of the structural and evolutionary relationships for all the entries in the Protein Data Bank.

SCOP was established by the careful manual approach of Dr. Alexei Murzin with published description of their structures, while CATH and FSSP are built more or less automatically from structural alignments. While the CATH and FSSP classifications use protein chains as the object of interest, SCOP breaks proteins into domains as a result of eliminating the problem of placing multi-domain proteins in the classification hierarchy.

SCOP has a complicated hierarchy with manually assigned domains classified into seven fold classes first, then classified into common folds, superfamilies and families. Each hierarchical level has the following explanation (Lindahl and Elofsson, 2000): proteins sharing family have a clear evolutionary relationship; those within a superfamily are probably of common evolutionary origin; while the fold level is characterised by major structure similarity.

2.2 Protein structure prediction

As stated in Section 2.1.1, the protein structure prediction problem is to predict the tertiary structure of a protein from its amino acid sequence. The

protein tertiary structure prediction problem could be regarded as transforming information. The input is a string of 20 alphabetic characters; each represents one of 20 types of amino acid. The desired output consists of the three XYZ coordinates in a three-dimensional fold shape correspond to each character. There is an increasing gap between the number of existing protein sequences and that of the known protein structures due to various genome sequencing projects around the world. In an attempt to identify protein structures quickly, researchers are trying to predict protein tertiary structures from their sequences by using existing biological knowledge and computational methods. There are three possible theoretical approaches to do this task, depending on the similarity of the query sequence to proteins of known structure, as shown in Figure 2.6.

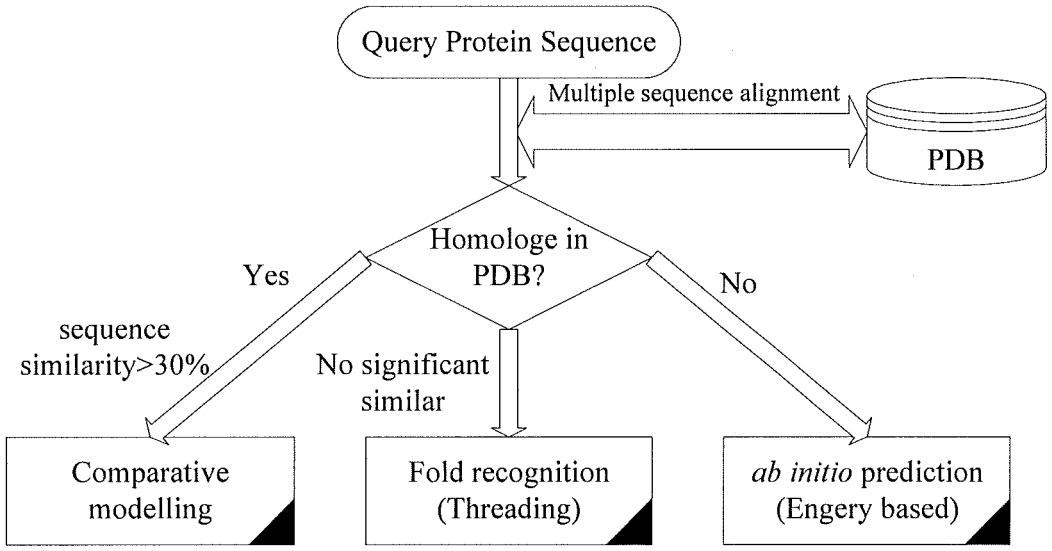


Figure 2.6 Methods for Protein Structure Prediction

Given a protein sequence of unknown structure, sequence alignment / multiple sequence alignment is applied through the known protein database (PDB) first, then, according to the homology of the query sequence and a known database, the comparative modelling, fold recognition (threading) or *ab initio* prediction methods could be used to predict an unknown protein.

2.2.1 Comparative modelling

Comparative modelling is currently the most reliable method for protein structure prediction. This method is also frequently referred to as "homology modeling". Successful predictions based on comparative models have been reviewed by Baker and Sali (2001). The technique is based on the observation that two proteins with very similar sequences tend to have similar backbone structures (Chothia and Lesk, 1986). So, it can only be applied when there are protein structure templates that share clear sequence similarity with the probe sequence. When the pairwise sequence identity between a probe and the template is higher than a certain threshold (e.g. 30%), the comparative modelling program can be used to build very accurate predictions for unknown proteins (Moult et al., 1995). The process of comparative modelling often includes building alignments between the templates and the probe sequence, copying the backbone structures from the templates according to the alignment, building a framework structure for the probe, adding loops and side chains, and refining and validation the model (Gibas and Jambeck, 2001).

Comparative modelling methods are highly developed. Even an automatic sever is capable of generating good models (Peitsch, 1996; Schwede et al., 2003). But with more remotely related template and probe, the building of loops and especially the alignment between the templates and the probe are still problematic. Protein fold recognition methods have been applied in comparative modelling to select structure templates and generate alignments between templates and probe sequence (Bates et al., 2001).

2.2.2 Sequence alignment and scoring matrices

Sequence alignment is the key step in protein structure prediction using the comparative modelling method. It is the most common way of

describing similarity between protein sequences. A dynamic Programming algorithm (such as Needleman-Wunsch (1970) and Smith-Waterman (1981)) is the most popular algorithm used in alignment programs. Needleman and Wunsch (1970) first introduced the dynamic programming algorithm into bioinformatics. With an assumption that the substitution of each residue is independent, the dynamic programming algorithm finds a single optimal alignment path given an amino acid substitution scoring matrix and a gap penalty function. In the alignment, the most similar segments of two sequences are aligned while the gap regions between them are minimized. Gotoh (1982) implemented a more efficient version. Smith and Waterman (1981) developed a slightly different algorithm. This algorithm detects the best alignment between subsequences of two sequences, which is often called local alignment, compared to the global alignment used in the Needleman-Wunsch algorithm (1970).

To align sequences by dynamic programming, it is often necessary to introduce relative insertion and deletions to attain a maximum matching of amino acids. Alignment gap penalties, which can also be viewed as a relative log likelihood of deletion or insertion, should be introduced in dynamic programming algorithms. The earliest gap penalty was a fixed one for each residue deleted or inserted, or a fixed penalty for a gap of any length (Needleman and Wunsch, 1970). The former often involved a large number of short insertions or deletions while the latter one could lead to extremely long gaps. Both were not biologically ideal. The most common form of gap penalty used now is the affine gap penalty, which can be written as $g = a + bn$, where g is the applied penalty, a and b are opening and extending parameters while n is the number of spaces in the gap. Often b is much closer to zero than a (Gotoh, 1982; Altschul and Erickson, 1986). Algorithms for constructing optimal global or local pairwise algorithms require $O(mn)$ time with these gap penalty functions, where m and n are lengths of sequences being compared. The $O(mn)$ means the

computing time of the algorithm is roughly proportional to the product of m and n. More complicated gap costs have been defined (e.g. Myers and Miller, 1988). For the class of concave gap penalties, we can still build optimal alignment algorithms that require only $O(mn)$ time. However, the implementation of such algorithms is more complex and error-prone. Almost all popular alignment programs use affine gap penalties.

Overall, dynamic programming algorithms are effective alignment methods. However, as the computing time of these algorithms is roughly proportional to the product of the length of two sequences, they are not very fast algorithms compared to most heuristic database searching algorithms.

The scoring system employed in dynamic programming is through the use of a substitution matrix. The substitution matrices have been generated from a variety of data sources on the basis that certain amino acids can substitute easily for another with similar physiochemical properties during evolution. Two kinds of matrices, PAM (Dayoff et al., 1978) and BLOSUM (Henikoff and Henikoff, 1992), are commonly used. The PAM (point accepted mutation) model of amino acid substitution was first introduced by Dayhoff and co-workers (1978). It has been a standard for protein sequence comparison for more than 20 years (Blake and Cohen, 2001). It is designed to score alignments between sequences that have diverged by a particular degree of evolutionary distance. In their Markov model, it was assumed that each mutational event was independent of previous events. A table of 20×20 mutation probabilities of amino acids at an evolution distance of 1 PAM (Point Accepted Mutation) was estimated using algorithms of sequences of closely related proteins. Substitution matrices appropriate for greater evolutionary distances can then be generated by repeated multiplication of 1 PAM matrix. The BLOSUM substitution matrices (Henikoff and Henikoff, 1992) have been constructed in a similar

fashion, but make use of a different strategy for estimating the target frequencies. The matrix values are built on much larger data sets than PAM matrices to find their conserved domain and involved distantly related sequences. Currently, BLOSUM is perhaps the most popular substitution matrix for pairwise alignment. It provides the foundation for a number of database search techniques including BLAST and PSI-BLAST (Altschul et al., 1997).

2.2.3 Threading/Fold recognition

The fold recognition method, which is also called protein threading, is used when there is an absence of a significantly sequence similarity with known structure. This is the case for the research in this thesis. A more detailed analysis on protein threading is given in Chapter 3. The basis of threading is the fact that there may only be a finite number of protein folds in nature (Govindarajan et al., 1999) and certain kinds of structure seem to be remarkably popular among apparently unrelated sequences (Chothia and Lesk, 1986; Rost, 1999; England et al., 2003). Although some new folds still can be obtained every year from structure determination experiments, the number of new folds is relatively small compared to the number of folds observed (Orengo et al., 2001). For a probe protein sequence with an unknown structure, it is likely that its fold has been seen, and proteins with similar structures are available in structural databases. At sequence identity levels beyond the twilight zone (<30%), comparative modelling methods are not reliable. And indeed, homologous sequences are often not found in present sequence database. So, fold recognition methods are designed to detect structure similarities and generate alignments.

A *threading* means a specific alignment between sequence and structure. Normally a scoring function is formulated in terms of the knowledge-based pseudo-energy potentials to evaluate protein sequence-structure

fitness. For the knowledge-based pseudo-energy potentials (for reviews, see Sippl, 1995; Jones and Thornton, 1996; Moult, 1997; Lazaridis and Karplus, 2000), quite often the statistical analysis of known protein structure is used to measure the free energy between the interaction of residues or atoms. The results of such analysis are commonly known as contact energies. In most cases, the knowledge-based potential is built on the assumption that pairwise contact between atoms or residues have independent contributions to the potential energy.

Both comparative modelling and fold recognition methods require appropriate templates to be present in the structure library. When there is no template that can be confidently identified, *ab initio* modelling methods can generate models without using full templates.

2.2.4 Ab initio modelling

Perhaps the most intuitive way of simulating protein folding is via molecular dynamic simulations with a physical potential function, because the physical interactions between atoms are clearly the driving force of protein folding. Obviously, the protein structure can be predicted via this approach without using structure templates. However, explicit representation of molecules and complex potential functions employed in such approaches require huge computing power. Also, accurate modelling of the potential function is a challenging problem itself. Only groups with a giant cluster of supercomputers like the IBM Blue Gene Project could be capable of performing such simulations for proteins of reasonable sizes.

With limited computing resources, most *ab initio* modelling methods work with greatly simplified models, which can be divided into two classes: lattice (Skolnick and Kolinski, 1991) and off-lattice models (Park and Levitt, 1995). By using these models, the complexity of the conformational

search can be sufficiently reduced because many details of protein 3D structures, including the coordinates of most atoms, are ignored. Once the representation of protein structure is specified, a scoring function must be developed to measure the quality of the different predicted models. The traditional one, which models the atomic force fields (Brooks et al., 1990), is not feasible with these reduced complexity representations. Many methods utilize scoring functions derived from the protein structure database that were adjusted to favour the native conformation over others. Such so-called knowledge-based pseudo-energy is also employed in the threading programs as mentioned above. With simplified representations and scoring functions, *ab initio* modelling programs search for near-native structures with Monte Carlo (Simons et al., 1997), simulated annealing and genetic algorithms (Jones, 2001).

In spite of encouraging recent improvements (Simons et al., 1999; Bonneau et al., 2001), most *ab initio* modelling methods are still limited to short protein sequences. Also, to build accurate models with *ab initio* methods remains a challenge.

2.3 Neural networks

2.3.1 Concepts of artificial neural networks

Artificial neural networks (ANNs) were first designed by McCulloch and Pitts (1943). They are computational models inspired by the modelling of the human brain. ANNs have a large number of highly interconnected processing elements (nodes) that usually operate in parallel and are configured in regular architectures. The simple processing units are often called artificial neurons or nodes. The connections between processing units are often called links. Each link is associated with a weight that

presents information being used by the net to solve a problem. The training of a NN is the procedure of finding the proper weights of the network.

The architecture of NNs is typically organized in layers; most applications have three normal types of layers—an input layer, a hidden layer and an output layer. The layer that receives input signals is called the input layer. The outputs of the network are generated from the output layer. Any layer between the input and output layers is called a hidden layer. Layers are made up of a set of interconnected nodes, as shown in Figure 2.7. It can be described as a directed graph in which each node i performs a transfer function f_i of the form

$$y_i = f_i(\sum_{j=1}^{n} \omega_{ij} x_j - \theta_i) \tag{2.1}$$

where y_i is the output of the node i, x_j is the jth input in the input layer to the node in the hidden layer, and ω_{ij} is the connection weight between nodes i and j. θ_i is the threshold (or bias) of the node. The f_i is called transfer function (or activation function) that can be linear or nonlinear function such as step function, hard limiter function, sigmoid function and Gaussian function, etc.

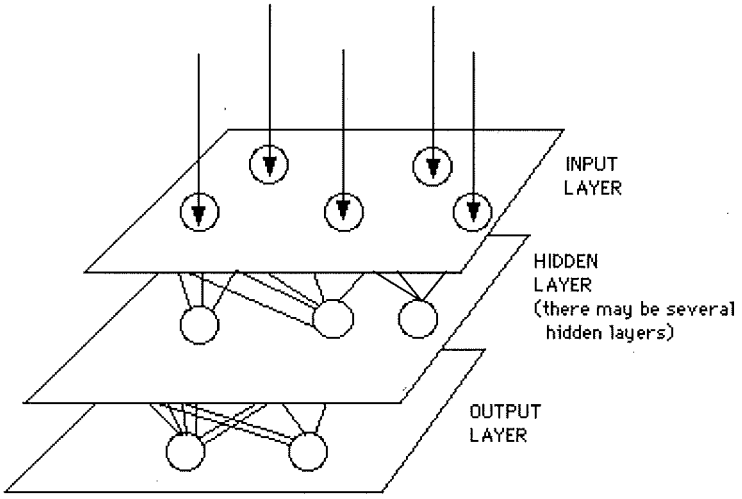


Figure 2.7 Architecture of ANNs

According to the connectivity of the neurons, ANNs can be divided into feed-forward and recurrent classes. The feed-forward networks have no output from processing elements being an input to another node in the same layer or in a preceding layer. When outputs can be directed back as inputs to the same layer, or preceding layer nodes, and have closed loops, the networks are named recurrent networks. The architecture of an ANN is determined by its topological structure, such as the layers, the overall connectivity and the transfer function of nodes in the network.

Most ANNs contain some form of learning rule that modifies the weights of the connections according to the presented input patterns. In general, learning rules are classified into two categories: supervised learning and unsupervised learning. In supervised learning, it is assumed that the correct "target" output values are known for each input pattern. The weights are usually obtained by minimizing some error functions, which measure the difference between the "target" and the values computed by the NNs. In unsupervised learning, there is no teacher to provide any feedback information. The network must discover for itself the patterns, features, regularities, corrections, or categories in the input data and code for them in the output. Although there are many different kinds of learning rules used by NNs, the most common one is a gradient descent-

based optimization algorithm called the back propagation learning rule, which is a supervised process that occurs with each epoch through a forward activation flow of outputs, and the backwards error propagation of weight adjustments.

Though the initial intent of ANNs was to explore and reproduce human information processing tasks such as speech, vision, and knowledge processing; ANNs also demonstrated their superior capability for classification and function approximation problems. This has great potential for solving complex problems such as system control, data compression, optimization problems, pattern recognition, and system identification. Recently, NNs have been widely applied in bioinformatics to solve complicated problems that are difficult to solve by traditional methods.

2.3.2 Backpropagation neural networks

The backpropagation neural networks (BPNNs) were proposed by Rumelhart et al. (1986). BPNNs are multilayer feed-forward networks combined with a back-propagation learning algorithm. BPNNs are currently the most general-purpose and commonly used NN paradigm, which achieve their generality because of the gradient-descent technique used to train the networks.

A feed-forward network has a layered structure. Each layer consists of units that receive their input from units in a layer directly below and send their output to units in a layer directly above the unit. There are no connections within a layer. As shown in Figure 2.8, the N_i inputs are fed into the first layer of $N_{h,1}$ hidden units. The activation of a hidden unit is a function f_i of the weighted inputs plus a bias. The output of the hidden units is distributed over the next layer of $N_{h,2}$ hidden units, until the last

layer of hidden units, from which the outputs are fed into a layer of N_o output units.

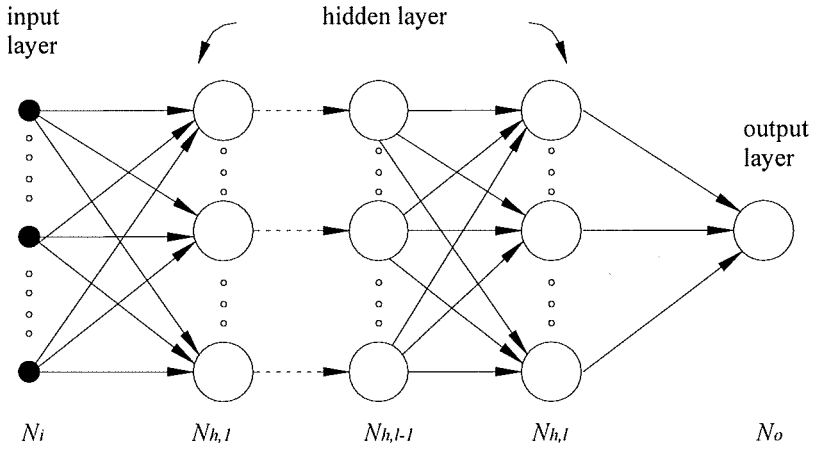


Figure 2.8 A multi-layer network with l layers of units

Although backpropagation can be applied to networks with any number of layers, it has been shown by Cybenko (1989) and Hartman et al. (1990) that one layer of hidden units suffices to approximate any function with finitely many discontinuities to arbitrary precision, provided the activation functions of the hidden units are non-linear.

In BPNNs, the central idea is that the errors for the units of the hidden layer are determined by backpropagating the errors of units of the output layer. This is called the backpropagation learning rule. Usually, a network is trained over a number of training pairs, which can be thought of as a set of ordered vector pairs $\{(I_1,d_1),(I_2,d_2),\cdots,(I_p,d_p)\}$ where each I_i represents an input vector and each d_i represents the output vector associated with the input vector I_i. The learning algorithm for the training of a BPNN is as follows:

1) Initialization: Decide the number of layers and neurons of BPNNs; initialize the weights and thresholds to some random values.

2) Training loop: Apply the *i*th input vector I_i to the input layer and specify the desired output vector d_i.

3) Forward propagation: At each node, calculate the weighted sum of the inputs and apply the appropriate activation function, calculate the actual output. The sigmoid function is a widely used activation function

$$X_j = \frac{1}{1+e^{-(x_j-\theta_j)}} \tag{2.2}$$

where x_j is the weighted sum of inputs coming to the jth node,

X_j is the output of the jth node,

θ_j is the threshold for the jth node.

4) Error back propagation: Propagate the errors backward to update the weights and adjust the weights by

$$W_{ij}(t+1) = W_{ij}(t) + \eta\delta_j x_j + \beta[W_{ij}(t) - W_{ij}(t-1)] \tag{2.3}$$

where W_{ij} is the weight from ith node to the jth node,

δ_j is the error at the jth node,

η is the learning rate,

β is the moment,

If j is an internal hidden layer node,

$$\delta^h{}_j = X_j(1-X_j)\sum_k(\delta_k W_{kj}) \tag{2.4}$$

where the summation is performed over all the nodes in the layers above the node j.

If j is an output layer node,

$$\delta^o{}_j = (d_j - y_j)f'(net^o_j) \tag{2.5}$$

5) Repeat step 2 through step 4 for as many epochs as it takes to reduce the sum squared error to a minimal value. If the training error is acceptable, terminate the training process.

6) Testing: Substitute the testing data into the network for testing. Calculate the error between the actual output value and the target output value of testing data.

7) Total error checking: If the error for testing data is acceptable, output the final weights; otherwise, adjust the architecture of the network and initiate the new training epoch by going to step 1.

After sufficient iterations of step 2 to step 5, the BPNNs can successfully learn to replicate all the training output vectors given any of the input patterns. Then the learning phase is stopped and the connection-weight values are frozen. The network is ready to be used in the specific application.

2.3.3 Neural networks in bioinformatics

It has been shown that NNs, as an automatic and intelligent learning technique, can be widely applied in bioinformatics and have had a lot of

success in this research area (Baldi and Brunak, 2001). The early application of NN algorithms to problems within the field of biological sequence analysis can be traced back to 1982, when the perceptron was applied to the prediction of ribosome binding sites based on amino acid sequence input (Stormo, et al., 1982). A perceptron without hidden units was able to generalize, and could find translational initiation sites within sequences that were not included in the training set.

The linear architecture of a perceptron is clearly insufficient for many sequence recognition tasks. The early pioneering work involved the use of BPNNs for protein structure prediction (e.g. Qian and Sejnowski, 1988; Holley and Karplus, 1989; Kneller et al., 1990) or DNA sequence discrimination (e.g. Lapedes et al., 1989; Uberbacher and Mural, 1991; Brunak, et al., 1991). As the field continues to develop, researchers have broadened the choices of NNs architecture and learning algorithms to solve a wider range of problems (reviewed by Wu, 1997).

In the early work of Qian and Sejnowski (1988), a NN is used to predict protein secondary structure. The input window is an optimal size of 13 amino acids. An orthogonal encoding scheme is used for input with size 21, corresponding to 20 amino acids and one for N- or C-terminal. Thus, the input layer has $13 \times 21 = 273$ units. The output layer of NNs has three units, with orthogonal encoding of the alpha-helix, the beta-sheet and the coil classes. The NN is trained to classify the residue located in the centre of the input window, into one of three secondary classes. The overall performance of their model reaches 64.3%. Most of the subsequent work on protein secondary prediction using NNs (Holley and Karplus, 1989; Kneller et al., 1990; Rost and Sander, 1993; Rost et al., 1994) has been based on the architecture of Qian and Sejnowski's (1988) model. The most significant performance improvement on protein secondary prediction compared to previous work has been done by Rost and Sander (1993), which resulted in the PHD prediction server (Rost et al., 1994). The PHD

method reached a performance level of 74%. The key features of the PHD approach are the use of multiple sequence alignments information (instead of single sequence) and a multi-level system instead of one NN. Recently, McGuffin et al. (2000) developed the PSI-PRED server for protein secondary structure prediction using NNs. They used an iterative approach to generate profiles as the improved input to the NN. These profiles were based on position-specific scoring matrices. It has been shown that using these profiles as input, significantly increased the accuracy of protein secondary structure prediction. To date, PSI-PRED (McGuffin et al., 2000) method is the best method for protein secondary structure prediction, reaching a performance of 77%.

A NN used for DNA sequence discrimination is treated as a pattern recognition model. The early work of NetGene (Brunak, et al., 1991) applied three BPNNs to predict acceptor and donor site positions in human genomic DNA sequences. Two NNs are used to predict local splice sites and joined with one NN to predict an exon. Snyder and Stormo (1995) developed the GeneParser system to predict gene structure using the combination of the NNs with dynamic programming. In the GeneParser, intron/exon and splice site indicators are weighted by a NN to approximate the log-likelihood that a sequence segment exactly represents an intron or exon. A dynamic programming algorithm is then applied to this log-likelihood to find the combination of introns and exons that maximizes the likelihood function. GeneParser precisely identifies 75% of the exons and shows as good a generalized performance as with the training set.

The earliest NN used in protein tertiary structure prediction was done by Bohr et al. (1990), who predicted the distance between amino acids of homologous protein sequences. Wilcox et al. (1991) and Xin et al. (1993) applied a large-scale NN to learn the PDB protein tertiary structures represented by 140×140 distant matrices. The produced network predicted

well the distance matrices from homologous sequence, but had a limited generalisation capability due to the small size of training set relative to the network size. Milik et al. (1995) developed a NN system for the evaluation of side-chain packing in protein structures. Instead of using protein sequence as input, the protein structure was represented by a side-chain-side-chain contact map. Recently, Lin et al. (2002) proposed a NN approach on protein threading score. A BPNN is trained to predict the compatibility of amino acid residue side chain with its tertiary structure environments.

Other applications of NN in bioinformatics include early sequence analysis studies (Hirst and Sternberg, 1992; Reczko and Suhai, 1994); transmembrane helices (Rost et al., 1996) and folding initiation sites (Compiani et al., 1998). Also NNs have been successfully applied to predict whether distances between pairs of amino acids are above or below a given variable threshold (Lund et al., 1997) and contact maps of proteins (Fariselli and Casadio, 1999; Fariselli, et al., 2001; Pollastri and Baldi, 2002).

As a well-known and well used method in bioinformatics, NN will continue to be a valuable tool in the analysis of the large volume of molecular sequence data being generated by the Human Genome Project. The advantage of ANNs are capable of learn and solve many real-world problems. They are very flexible and can alter their internal curve-fitting function to handle discrete-valued and vector-valued functions from different examples. They are very well suited to the "noisy" bimolecular data. That is why they have gained a lot success in the application of bioinformation.

2.4 Support vector machines

Support vector machines (SVMs) are a new generation of machine learning algorithm (Boser et al., 1992; Vapnik, 1998), which have received much consideration because of their superior performance in a wide variety of application domains such as handwriting recognition, object recognition, speaker identification, face detection and text categorization (Cristianini and Shawe-Taylor, 2000). Generally, the SVMs are universal approximators that can be used to learn a variety of representations from a set of positively and negatively labeled training samples. A complete description to the theory of the SVMs could be found in Vapnik's book (Vapnik, 1998). Here a brief introduction of basic ideas behind the SVMs is described below.

2.4.1 Basic Idea of SVM

A SVM is a margin classifier. It attempts to construct a separating hyperplane between training data separating class members (positive examples) from non-members (negative examples). Using this separating hyperplane, an unknown sample can be identified as a member or non-member of the class based on whether it is on the member or non-member side of the hyperplane. Unfortunately, for most real-world problems it is impossible to construct a separating hyperplane, as the input space of Figure 2.9 demonstrates. The SVM solves this inseparability problem by mapping data from its original k-dimensional space into a higher-dimensional space and defines a separating hyperplane there. The original k-dimensional data space is called the input space and the higher-dimensional space is called the feature space, as shown in Figure 2.9.

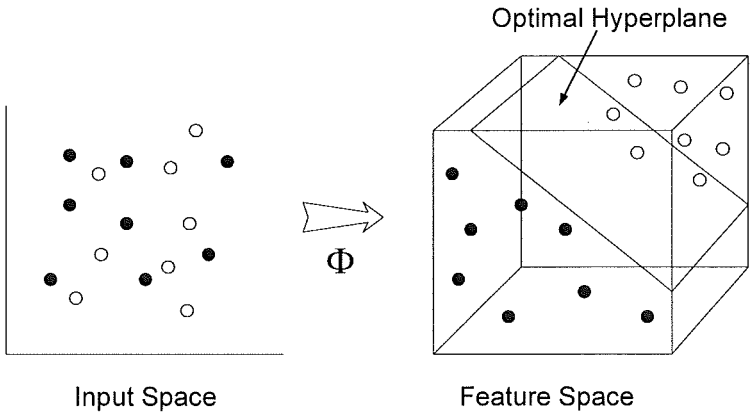


Figure 2.9 An illustration of SVM: Given a nonlinear classification problem in the input space, the SVM method defines a mapping Φ, and constructs the optimal separating hyperplane in the higher-dimensional feature space. Black and white circles indicate positive and negative samples to be classified.

2.4.2 SVM Mathematics

In this section, linear learning machines are introduced first, which are the foundation of SVMs, and then the non-linear cases are described.

2.4.2.1 The linear separable case

As shown in Figure 2.10, giving a weight vector $\vec{w}$ and a threshold b, there exists a separating hyperplane whose function is $\vec{w} \cdot \vec{x} + b = 0$, which implies:

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1, i = 1, 2, \cdots, n \qquad (2.6)$$

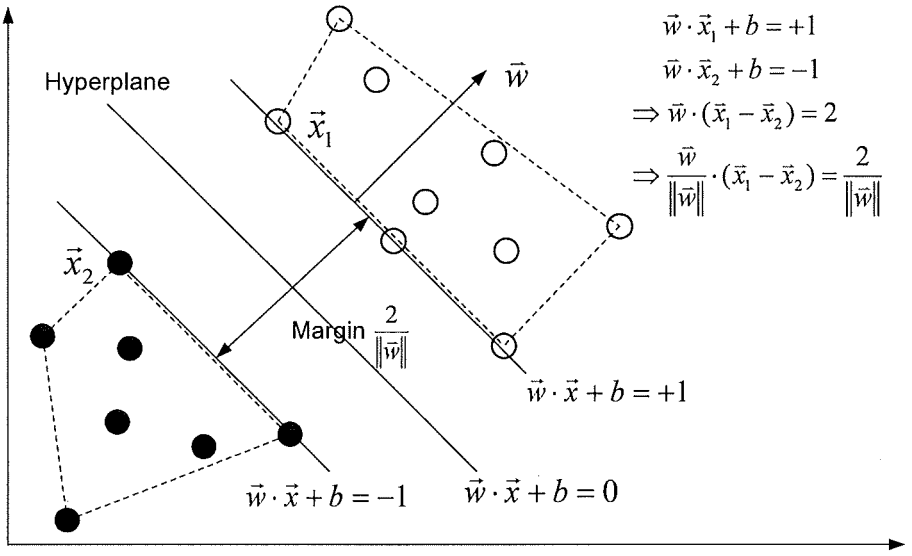


Figure 2.10 Definition of hyper-plane and margin. The black and white circles represent samples of class -1 and class +1, respectively. The optimal hyperplane is the orthogonal to the shortest line connecting to the convex hulls of the two classes (dotted lines), and inserts it half way. The margin, measured perpendicular to the hyperplane, equals $\frac{2}{\|\vec{w}\|}$.

For each group of training data, there exist a number of hyper-planes. The classification objective of the SVM is to determine an optimal weight and an optimal bias such that the selected hyperplane separates the training data with maximum margin. To maximize the margin, $\|\vec{w}\|$ needs to be minimized subject to the constraint (formula 2.6). By introducing Largrange multipliers α_i, the SVM training procedure amounts to solve a convex Quadratic Programming (QP) problem. It turns out, due to the nature of the QP problem, that only those points situated $\alpha_i > 0$ are called support vectors, $x_i, i = 1, 2, \cdots, N_s$. Thus, given an input training samples $\{\vec{x}_1, \vec{x}_2, \cdots \vec{x}_i, \cdots, \vec{x}_n\} \in \Re^d$ with known class labels $\{y_1, y_2, \cdots, y_i, \cdots, y_n\}, y_i \in \{+1, -1\}$, a new data point $\vec{x}$ could be assigned a label by the trained SVM according to the decision function:

$$f(\vec{x}) = \text{sgn}[\sum_{i=1}^{N_s} \alpha_i y_i \cdot \vec{x}_i \cdot \vec{x} + b] \qquad (2.7)$$

2.4.2.2 The non-separable case

As real-world problems are usually non-linear, the following approach has been introduced into SVM to deal with these problems.

- "soft margin" technique

In this case, some training examples are allowed to fall on the wrong side of the hyperplane. By introducing slack variables $\xi_i > 0, i = 1, \cdots, n$, a relaxed separation constraint is given as:

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 - \xi_i, i = 1, 2, \cdots, n \tag{2.8}$$

and the optimal separating hyperplane can be found by minimizing

$$\frac{\|\vec{w}\|^2}{2} + C\sum_{i=1}^{n} \xi_i \tag{2.9}$$

where C is a regularization parameter used to decide a tread-off between the training error and the margin.

- "kernel function" technique

The input vector $\vec{x}$ from the input space $\Re^d$ is mapped into a higher dimensional feature space $\hbar$ by a nonlinear kernel function. The motivation for mapping the data into high-dimensional feature space is that linear decision boundaries constructed in the high-dimensional feature space correspond to non-linear decision boundaries in the input space. The form of the decision function is:

$$f(\vec{x}) = \text{sgn}[\sum_{i=1}^{N_s} \alpha_i y_i k(\vec{x}_i \cdot \vec{x}) + b] \tag{2.10}$$

where

$$k(\vec{x}_i \cdot \vec{x}) = \langle \Phi(\vec{x}_i), \Phi(\vec{x}) \rangle \tag{2.11}$$

is the kernel function. $\Phi(\vec{x})$ is a nonlinear mapping function from the input space to feature space ($\Phi : \Re^d \mapsto \hbar$). The idea of the kernel function is to enable operations to be performed in the input space rather than the potentially high dimensional feature space. Thus, the mapping function Φ need not be explicitly defined because the algorithm only requires the evaluation of the inner product in (2.11). Several typical kernel functions are listed:

$$k(\vec{x}_i \cdot \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j + 1)^d \tag{2.12}$$

$$k(\vec{x}_i \cdot \vec{x}_j) = \exp(-\gamma \left\| \vec{x}_i - \vec{x}_j \right\|^2) \tag{2.13}$$

$$k(\vec{x}_i \cdot \vec{x}_j) = \tanh(k(\vec{x}_i \cdot \vec{x}_j) + \theta) \tag{2.14}$$

$$k(\vec{x}_i \cdot \vec{x}_j) = \frac{1}{\sqrt{\left\| \vec{x}_i - \vec{x}_j \right\|^2 + c^2}} \tag{2.15}$$

Equation (2.12) is the polynomial kernel function of degree d which revert to the linear function when $d = 1$. Equation (2.13) is the radial basic function (RBF) kernel with one parameter γ. Equation (2.14) is the sigmoid kernel and equation (2.15) is the inverse multi-quadric kernel.

2.4.3 Properties of SVM

A SVM model, an efficient classifier, has a number of properties:

- It is based on statistical learning theory. Its unique ability to develop a model with superior generalization capabilities makes it

the most suitable tool among various supervised learning algorithms when the number of input features is large compared to the number of training samples.

- It is practical as it reduces to a quadratic programming problem with a unique solution. The solution of the QP problem is globally optimised while some other training algorithms only guarantee finding a local minimum.

- It can effectively avoid over-fitting by choosing the maximum margin separating hyperplane from among the many that can separate the positive from negative examples in the feature space.

- It contains a number of heuristic algorithms as a special case. That is, by the choice of different kernel functions, different architectures could be obtained. The dot product represented by kernel functions in feature space avoids the "curse of dimensionality".

- It can automatically identify a small subset from the input samples as support vectors, thus avoid the computational burden.

2.4.4 Support Vector Machines Application in Bioinformatics

The SVM approaches in bioinformatics include the recognition of translation start sites in DNA (Zien et al., 2000), the gene and tissue classifications from microarray expression data (Brown et al., 2000; Fuery et al., 2000; Guyon et al., 2002; Vert and Kanehisa, 2003), protein remote homology detection (Jaakkola et al., 1999a; Liao and Noble, 2002; Leslie et al., 2003), protein fold recognition (Ding and Dubchak, 2001), protein secondary structure prediction (Hua and Sun, 2001b), protein subcellular localization prediction (Park and Kanehisa, 2003) and peptide

identification from mass spectrometry data (Anderson et al., 2003). As a new learning technique, the broad applications in bioinformation by SVM are due to the efficient features and good generalisation performance of SVM. Some successful applications are listed below.

2.4.4.1 Gene classification

The first application of SVMs to microarray data involved the classification of yeast genes into functional categories (Brown et al., 2000). A total of 2467 genes of the budding yeast *S. cerevisiae* were represented by a 79 dimensional gene expression vector, and classified according to six functional classes. The SVM yields very good performance on this task in comparison with some selected traditional machine learning techniques. This application successfully used the SVM both for the task of classifying unseen genes and for cleaning existing datasets by identifying genes in the training sets that had been mislabeled.

Pavlidis et al. (2001) applied a SVM to infer gene functional classification from heterogeneous data sets consisting of DNA microarray expression measurements and phylogenetic profiles from whole-genome sequence comparisions. This work assumes that genes with similar switching mechanisms are likely to operate in response to same environmental stimulation and hence are likely to have similar or related function roles.

2.4.4.2 Tissue classification

Mukherjee et al., (1999) first demonstrated the application of the SVM to a tissue classification task. Because of the high dimensionality of the examples, a linear kernel is applied. Mukherjee et al., (1999) also describe a technique for assigning confidence values to the SVM prediction. The method assumes that the probability of a particular class, given a particular example, is approximately equal to the probability of the class

given the corresponding SVM's discriminant value. Discriminant values are estimated using leave-one-out cross-validation, and their distribution is estimated using a SVM-based, non-parametric density estimation algorithm (Mukherjee and Vapnik, 1999).

In work carried out concurrently, Moler et al., (2000) describe the application of SVMs to the recognition of colon cancer tissues. This work describes a general, modular framework for the analysis of gene expression data, including generative Bayesian methods for unsupervised and supervised learning, and the SVM for discriminative supervised learning.

In a similar set of experiments, Furey et al., (2000) apply linear SVMs with feature selection to three cancer data sets. The SVM successfully identified a mislabeled sample in the ovarian set, and is able to produce a perfect classification.

Lee and Lee (2003) extended these two classes classification into multiple cancer type by introducing multicategory SVMs. The approach was tested on the AML (acute myeloid leukemia)/ALL (acute lymphoblastic leukemia) and small round blue cell tumours and showed perfect performance.

Segal et al., (2003a) use the SVM to develop a genome-based classification scheme for clear cell sarcoma. This type of tumor displays characteristics of both soft tissue carcoma and melanoma. Using 256 genes selected via a t-test, a linear SVM is trained to recognize the distinction between melanoma and soft tissue sarcoma. In leave-one-out setting, the classifier correctly classifies 75 out of 76 examples. Related work has also been carried out by Segal et al. (2003b). This time, a SVM is used to investigate the complex histopathology of adult soft tissue sarcomas.

By considering the gene-gene correlations occurring in the gene data, Guyon et al., (2002) proposed a SVM-based learning method, called SVM recursive feature elimination (SVM-RFE). This SVM-RFE algorithm is tested on the AML/ALL and colon cancer data sets.

Besides the application of SVMs on tissue classification above, Su et al. (2003) describe a tool called RankGene that produces gene ranking. One of the ranking metrics available in RankGene is the discriminant of a one-dimensional SVM trained on a given gene. Yeang et al. (2001) addressed many tissue classification problems with SVMs.

2.4.4.3 DNA and RNA

Zien et al. (2000) compare SVMs to a previously described NN approach on the recognition of translation start sites in DNA. A fixed-length window of DNA is encoded in redundant binary form, and the SVM and NN are trained on the resulting vectors. Using a simple polynomial kernel function, the SVM improves upon the NN's error rate (from 15.4% down to 13.2%). A similar application is described by Degroeve et al. (2002). Here, rather than recognizing the starts of gene, the SVM learns to recognize the starts of introns.

In contrast with the two methods above, which aim to recognizing specific sites in a DNA, Carter et al. (2001) have demonstrated the application of SVMs to the problem of recognizing functional RNAs in genomic DNA. Functional RNAs (fRNAs) are RNA molecules that have a functional role in the cell and do not code for a protein molecule. In the approach used by Carter et al. (2001), the SVM performs well and slightly better compared to a NN.

2.4.4.4 Protein analysis

SVMs have been less prevalent in protein analysis compared with NNs (Baldi and Brunak, 2001). The first application of SVMs on proteins was proposed by Jaakkola et al. (1999a; 1999b) and the described algorithm was called SVM-Fisher. The SVM-Fisher method couples an iterative HMM (Hidden Markov Model) training scheme with the SVM. For any given family of related proteins, the HMM provides a kernel function. By combining HMMs and SVMs, SVM-Fisher offers an interpretable model with an excellent recognition performance. Subsequent work by Karchin et al. (2002) demonstrates the successful application of the SVM-Fisher methodology to the recognition of large, pharmaceutically important class of protein, the G-protein coupled receptors.

For protein remote homology, Ding and Dubchak (2001) define a composition based kernel function that characterizes a given protein via the frequency with which various amino acids occur therein. In this work, each protein is characterized by a simple vector of letter frequencies. Each protein sequence is represented via six different alphabets, corresponding to amino acids, predicted secondary structure, hydrophobic, normalized van der Waals volume, polarity and polarizability. A single protein is represented by the letter frequencies across each of these alphabets, for a total of 125 features. Ding and Dubchak introduce a method called the unique one-vs-others method, which performs additional SVM optimizations in order to sort out disagreements among SVMs training using the standard, one-vs-others method. They show that their method leads to significant improvement in the test set accuracy. The work also shows that a SVM outperforms a similarity trained NN on this task.

A similar SVM model is used by Cai et al. (2001) to recognize broad structure classes of proteins (all α, all β, α/β and $\alpha+\beta$). On this task,

the SVM also shows a better discrimination performance than a NN method.

Hua and Sun (2001a) use SVMs to perform protein classification with respect to subcellular localization. In this work, the SVM is shown to produce more accurate classifications than competing methods, including a NN.

Zavaljevski and Reifman (2002) describe the application of a SVM to a clinically important, binary protein classification problem. The class of human antibody light chain proteins is large and is implicated in several types of plasma cell diseases. In particular, Zavaljevski and Reifman use SVMs to classify the k family of human antibody light chains into benign or pathogenic categories. The resulting classifier yields an accuracy of around 80%, measured using leave-one-out cross-validation, which compares favourably with the error rate of human experts.

In addition, Hua and Sun (2001b) have demonstrated how to predict protein secondary structure with SVMs. The resulting classifier achieves a pre-residue accuracy of 73.5% on a standard data set, which is comparable to existing methods based upon NNs.

Koike and Takagi (2004) use SVMs to identify the protein-protein interaction sites, which is essential for the mutant design and prediction of protein-protein networks. The interaction sites of residue units were predicted using profiles of sequentially/spatially neighbouring residues, plus additional information. This prediction performance appeared to be slightly higher than a previously reported study.

2.5 Summary

Artificial intelligence techniques, especially NNs and SVMs, have been successfully used in interpreting and analyzing the large volume of biological data (Baldi and Brunak, 2001; Hua and Sun, 2001a; Ding and Dubchak, 2001). This research is focused on protein threading by artificial intelligence techniques. The related concepts and theory to this research project have been reported in this chapter.

Firstly, an overview of the spatial conformations of protein with the focus on protein primary and tertiary structure is given. With the understanding of protein structure, three possible protein prediction methods are summarized. They are comparative modelling, fold recognition (threading) and *ab initio* prediction method. Fold recognition method, also called protein threading, is used when the target sequence has an absence of a significantly sequence similarity and there is no homologous proteins with known structures. Currently, protein threading has become an important research area.

Secondly, from the reviewed applications of NNs and SVMs on bioinformatics, it has been shown that both NNs and SVMs have been successfully used for the analysis of biological problems. They learn a pattern based on training data and predict on future data. They are very well suited for domains with an abundance of data and lack of clear theory, which is precisely the case in protein structure prediction problem.

In this thesis, a framework for protein threading is proposed by using NNs and SVMs. With the above background introduction, the theory of protein threading will be analyzed in the following Chapter. A literature review will be given on the most recent and best performance threading servers.

CHAPTER 3 THREADING ANALYSIS AND RESEARCH FRAMEWORK

Protein threading (fold recognition) is proposed for those target sequences that have the same fold as some proteins with known three-dimensional structures but do not have homologous proteins with known structures. Protein threading makes a structure prediction through placing (aligning) the residues of the target sequence sequentially to the positions in the template to see whether the target can have the same fold as the template or not. Gaps are allowed in the alignment to some extent. Not all sequence residues are aligned to a template position and not all template positions are aligned by a sequence residue.

In addition to the introduction of threading in Section 2.3.3, a more detailed knowledge of the threading method and literature review are given in this chapter. The current research work on protein threading is reviewed in Sections 3.2 and 3.3, and Section 3.4 presents the research framework.

3.1 Analysing the Threading Program

Threading, which is also referred to as fold recognition, attempts to assign folds to sequences which show very low sequence identity to a known structure. Figure 3.1 shows a simple outline of how threading methods generally work. The amino acid sequence of a query protein (target protein) is examined for compatibility with the structural core (α helix, β strand and other structural element) of a known protein structure against a library of fold templates. If a reasonable degree of compatibility (with the highest similarity score or the lowest energy potential) is found with a given structural core, the protein is predicted to fold into a similar three-dimensional configuration. There are two common methods for determining whether or not a given protein sequence is compatible with a known structural core. They are the **structural profile method** and the **contact potential method**.

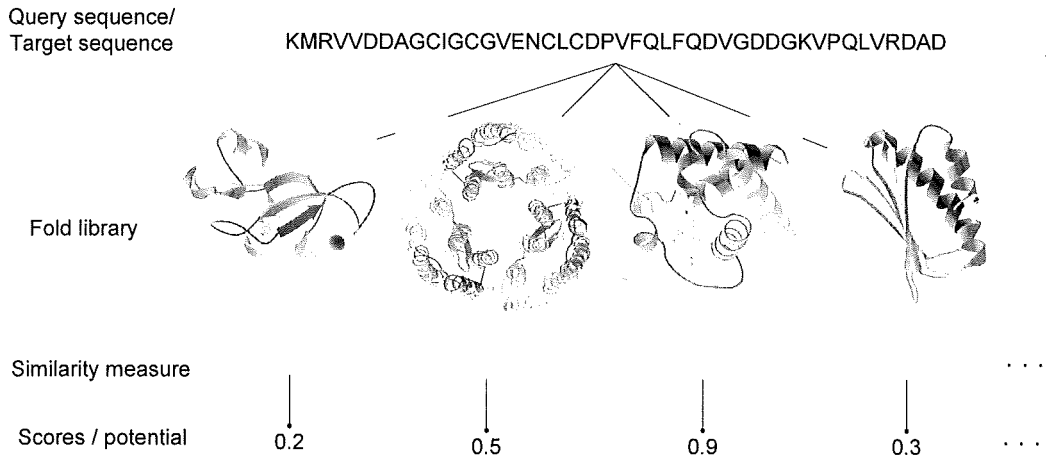


Figure 3.1 A simple outline of protein threading procedure.

The structural profile method was firstly developed by Bowie et al. (1991). By describing the structural environment of each amino acid residue in the structure templates, they attempted to match the templates with sequences using the preferences of amino acids in different environments.

The environment was described in terms of local secondary structure, solvent exposure and degree of burial by polar rather than a-polar atoms. On the basis of these environment descriptions, each amino acid is assessed into one of each group. For example, an amino acid with a hydrophobic side chain may fit best into the structure of buried group at that position. The query sequence is then aligned with a series of such environmentally defined positions in the structure to see whether a series of amino acids in the sequence can be aligned with the assigned structural environments of a template protein. The procedure is then repeated for each template in the structural database, and the best matches of the query sequence to the template are identified. It is assumed that the residue structural environment is more conserved than the residue itself, so the method can detect more remote relationships than pure sequence based methods. The method has been improved by much research (Rost, 1995; Russell et al., 1996; Rice and Eisenberg, 1997). Because of the improvements in secondary structure prediction accuracy, the predicted secondary structure and residue exposures of probe sequence were also included into the scoring scheme (Fischer and Eisenberg, 1996; Mallick et al., 2002).

The contact potential method was firstly introduced by Jones et al. (1992). The method was built upon the threading concept of Bowie et al. (1991), but differed from the method of Bowie et al. (1991) in that it considered the detailed network of pairwise interactions between individual residues rather than just assigning them to a basic environmental class (Jones and Hadley, 2000). In their method, a given protein fold is modeled as a network of pairwise interactions between amino acid residues. A sequence is matched to a structure by considering pairwise interactions, rather than local residue structural environments only. By including non-local interactions, threading models aim to detect even more remote relationships between templates and probes. However, the inclusion of

non-local interactions prohibits the use of the classic dynamic programming algorithm, because the assumption of independence in the dynamic programming algorithm is no longer valid. Thus, an iterative approach, which was developed for protein structure alignment (Taylor and Orengo, 1989), was introduced for making structure-sequence alignments. Recursive dynamic programming (Thiele et al., 1999), Gibbs sampling algorithm (Bryant, 1996), and other heuristic algorithms (Huber et al., 1999; Xu and Xu, 2000) have been developed to generate alignments in more efficient ways.

In Sections 3.2 and 3.3, the existing works on both approaches are reviewed and discussed.

3.1.1 Why Threading?

After a long period of evolution, the sequences of the proteins are extremely varied, whereas the three-dimensional structures are much more restricted because a fraction of residue exchange does not affect the stability of structures. The number of unique structural folds is fairly small (approximately 1000 folds; Govindarajan et al., 1999; Orengo et al., 2001). That means, amino acid types have different preferences for occupying different structural environments, and they might have distinct preferences for side-chain contact. Therefore, it is possible to quantify these interaction preferences of amino acids and produce a score function. This score function could identify the extent of those amino acids from the sequence located in preferred environments and adjacent to preferred neighbours. By doing this, the sequence can be threaded into the structure by searching for the best alignment that optimizes the score function. It is estimated that up to 70% of new protein sequences, their structure have a similar fold in the PDB, from which a suitable model could be constructed (Jones and Hadley, 2000).

It is generally assumed that comparative modelling methods are only able to recognize closely related sequences (Jones, 1999). When there are no obvious sequence similarities between a target and a template, a threading program can be adopted instead. Currently, threading has become a popular technique for protein structure prediction and achieved some success. For example, the threading program PROSPECT (Xu et al., 2001) performed the best in the CASP4 (Critical Assessment of Techniques for Protein Structure Prediction) competition.

3.1.2 Threading overview

For a threading program, there are some common elements:

A sequence of interest and a library of templates or known structure with all known folds (coordinate and angle).

To construct a library of potential core folds or structural templates is one of the basic components for getting good performance for a practical threading program. If the library is too large, the threading calculation could be very slow. If the library is too small, the correct template may not be included in the library and lead to the wrong conclusion of discovering a new fold. The members in the library usually consist only of abstractions of known structures, which is annotated with environmental features, such as, spatial adjacencies and distance between amino acids, solvent ability of amino acid itself, backbone parameters, secondary structure, and so on.

Arranging the sequence on each location of template, searching the best fit by some score with gaps and insertions.

Each distinct threading is assigned a score by a specific score function. The score function usually describes the degree of sequence-structure

compatibility between sequence amino acids and their corresponding positions in the core template as indicated by the alignment. It should have the ability to evaluate individual sequence residue preferences for the structural environment. For example, such a function should reflect the fact that a hydrophobic sequence residue may be more likely to occur in a buried structural environment than in an exposed one.

The alignment of the sequence to a given template usually is selected by searching the best alignment under the score function. It is an optimization problem from the viewpoint of mathematics. There are several possible approaches including dynamic programming (Needleman and Wunsch, 1970; Smith and Waterman, 1981), double dynamic programming (Jones et al., 1992) and Monte Carlo (Bryant and Altschul, 1995) / simulated annealing method (Kirkpatrick et al., 1983).

Going through the entire library, collecting the scores for all the candidate models, taking the best scoring one as the prediction model.

For each threading application, an optimal alignment between a query sequence and each structure in the template library needs to be calculated. Then a decision needs to be made on which sequence-structure alignment is the correct fold recognition. Until now, it is still a highly challenging and unsolved problem (Xu, et al., 2002).

3.2 Existing threading programs - structural profile approach

Within the structural profile method, it is assumed that if the query protein folds the same way as a target structure, the environments of the amino acids will be in the same linear order as they are in the target. Thus,

the structure of protein is encoded as a sequence of residue environments. A profile that describes the 3D environment of the template structure is made for each fold in the library. These profiles are used to score the compatibility between the query sequence and the representative fold. Various dynamic programming algorithms (global, semi-global, local and global-local) are used to identify an optimal, best-scoring alignment between sequence and profile. Bowie et al. (1991) firstly developed the structural profile method. Since then, many threading programs have been designed follow the structural profile approach. Some of the most recent well-designed threading programs are reviewed here.

3.2.1 UCLA-DOE fold server

The former UCLA-DOE fold server (http://fold.doe-mbi.ucla.edu/) is a fold-recognition server using 3D profiles and secondary structure prediction method as described by Fischer and Eisenberg (1996). The current server imported some new techniques like PSI-PRED (McGuffin et al., 2000) and DASEY (Mallick et al, 2002) to assign a structure for query sequence. DASEY (Directional Atomic Solvation energY) is an atom-based threading method and will be introduced in Section 3.2.5 below.

In the original approach, Fischer and Eisenberg (1996) defined a new sequence-structure compatibility function. The function combines the previously developed amino acid to structure compatibility scores (e.g. 3D-1D scores of Bowie et al., 1991) with the sequence-derived properties of the probe sequence. Various combined compatibility functions have been tested in their work. They are the combination of four different substitution tables (Henikoff and Henikoff, 1992; Gonnet et al., 1992), Bowie's 3D-1D scores plus the sequence-derived properties of the probe sequence. The sequence-derived properties of the probe sequence, such as the predicted secondary structure and solvent accessibility, are

demonstrated to be useful in protein fold recognition. The query sequence with the derived properties is aligned to the template using the global-local alignment algorithm. The predicted secondary structure and solvent accessibility are obtained from the PHD server (Rost et al., 1994). In the current UCLA-DOE server, PSI-PRED (McGuffin et al., 2000) is used for the secondary structure prediction.

3.2.2 GenTHREADER

GenTHREADER (Jones, 1999) is a fast and powerful protein fold recognition program. The method can be divided into three stages. First, a sequence-structure alignment is generated by global-local dynamic programming alignment algorithm. Alternatively, a query sequence profile is built to align with the structure profile in fold library. The one with the highest scoring alignment is taken as the preferred one. Then the alignment is evaluated by statistical potentials derived from THREADER program (Jones et al., 1992). The pairwise potential of mean force and the solvation potential are used by GenTHREADER model. Finally, the GenTHREADER uses a simple feedforward NN to produce a single score measuring the confidence in the prediction. The NN is trained with six scores. They are: sequence alignment score, number of aligned residues, length of query and template protein sequence, pairwise energy sum and solvation energy sum. The output of the NN is the binary CATH (Orengo et al., 1997) relationship. That is, pairs of proteins are randomly selected from CATH database. If the two domains of a pair are from the same topology family in CATH, the target value of NN is set to 1, otherwise to 0. The GenTHREADER server can be accessed from the link of http://bioinf.cs.ucl.ac.uk/psipred/.

3.2.3 3D-PSSM (three-dimensional position-specific scoring matrix)

3D-PSSM (Kelley et al., 2000) is a program to recognize remote protein sequence homologues. It implements the combination of multiple sequence profiles with structural-based profiles. Three different alignments between a target sequence and a template by using different scoring functions and different alignment policies are calculated. The alignment with the highest standardized score is taken as the final result. In each alignment, the scoring function contains secondary structure information, solvent accessibility, 1D-PSSM and 3D-PSSM information, as well as a gap penalty. The 1D-PSSM is generated from the multiple sequence alignment of a family of proteins as implemented in PSI-BLAST. The 3D-PSSM is constructed from the structural alignment program SAP (Orengo et al., 1992) for a superfamily of proteins. The three different alignments are: the target sequence is aligned to the 1D-PSSM of the template; the target sequence is aligned to the 3D-PSSM of the template; and the template sequence is reversely aligned to the 1D-PSSM of the target sequence. Since all alignments are involved with only sequence to profile alignment, a dynamic programming algorithm can be used to search for the optimal alignment.

The 3D-PSSM program is the first contemporary method to explicitly use information from structural alignments to aid protein fold recognition. The 3D-PSSM sever is available at http://www.sbg.bio.ic.ac.uk/~3dpssm/, where a user may submit a query sequence to be scanned against the 3D-PSSM database. The server performs a secondary structure prediction, and permits interactive viewing of alignment, and automatically generated preliminary models.

3.2.4 FUGUE

FUGUE (Shi et al., 2001) aims at the recognition of distant homologous by sequence-structure comparison. It aligns multiple sequences to multiple structure profiles. The multiple sequence alignment is generated by PSI-BLAST. The structural profile is derived from HOMSTRAD (Mizuguchi et al., 1998), which is a database of protein structure alignments for homologous families. At each template position, the structure profile is an environment substitution table. Three features are selected to describe local environment of a known protein structure. They are: main-chain conformation and secondary structure, solvent accessibility and hydrogen bonding status. The environment-specific substitution tables are built with the three groups of features. In the FUGUE program, a position-dependent gap penalty is used in the scoring function. At each position, the gap penalty is dependent on the solvent accessibility at this position, and its position relative to the secondary structure elements. FUGUE used the global-local algorithm to align a sequence-structure pair when they greatly differ in length and use the global algorithm in other case.

FUGUE is one of the better performing threading programs currently available (http://www-cryst.bioc.cam.ac.uk/~fugue/). Given a query sequence (or a sequence alignment), FUGUE scans a database of structural profiles, calculates the sequence-structure compatibility scores and produces a list of potential homologues and alignments.

3.2.5 DASEY

DASEY (Directional Atomic Solvation EnergY; Mallick et al., 2002) is an atom-based threading program. It extends the residue environmental definition introduced by Bowie et al. (1991). The environment of each protein position is encoded as the distribution of nonhydrogen atom types

along four tetrahedral directions from the α -carbon of the residue in that position. DASEY adopted the previous work of Sequence Derived Properties (SDP) used in UCLA-DOE server to mimic the fold assignment process. That is, the preference of a query sequence residue and its predicted secondary structure are computed for scoring function. DASEY has been demonstrated to perform better than some earlier procedures due to the atom-based more elaborate structure environmental description.

3.2.6 WURST

WURST (Torda et al., 2004) is a protein threading program with an emphasis on high quality sequence to structure alignments. The server is available at http://www.zbh.uni-hamburg.de/wurst/. First, a conservative sequence profile is built for the target sequence using PSI-BLAST. Then the sequence profile is aligned to about 9765 PDB template structures using local dynamic programming alignment algorithm. A sequence to structure score is calculated at each sequence position. Three-dimensional protein models, with side-chain only, are built from all alignments and evaluated using a more expensive quasi-energy function. The gap penalty is based on the distances within the model. The final score associated with each model is the combination of the alignment score, rescored model and gap penalties. Currently the final ranking of structure and confidence measurement employed in WURST is the same as the one used in GenTHREADER (Jones, 1999).

3.2.7 SPARKS

SPARKS (Sequence, secondary structure Profile And Residue-level Knowledge-based energy Score; Zhou and Zhou, 2004) is a threading program with an elaborately designed knowledge-based potential

function. Unlike 3D-PSSM (Kelley et al., 2000) that the score function only takes the secondary structure and solvent exposure into account, SPARKS developed a profile-energy score with a torsion-angle term for backbone interaction, a combined buried surface term and a contact-energy term for residue-residue and residue-solvent interactions. SPARKS also combines the elaborately designed score with the sequence profiles generated from PSI-BLAST (Altschul et al., 1997) and the secondary structure information predicted from PSI-PRED (McGuffin et al., 2000) for fold recognition. A global-local dynamic programming algorithm is employed to align query sequence profile to structural template profile in the fold library. SPARKS gains some improvement on sensitivity and alignment accuracy compared to several other methods mentioned in their paper. The improvement may due to the sophisticated backbone and side interactions imported into the score function.

3.3 Existing threading programs - contact potential approach

Typically, the contact potential method models interactions in a protein structure as sum over pairwise interactions. The formalization of the problem is:

Given a template structure T with positions $t_1, t_2, \cdots t_n$, and a query sequence S with amino acids $a_1, a_2, \cdots, a_n$, to find an $A = A(1), A(2), \cdots, A(n)$ (where $1 \le A(1) < A(2) < \cdots < A(n) \le m$ and $A(i)$ indicates the index of amino acid from S that occupies t_i) such that $\sum_{i=1}^{n}\sum_{j=1}^{n} score(i, j, a_{A(i)}a_{A(j)})$ is maximized.

The method was first introduced by Jones et al. (1992). In the residue-residue contact potential method, the number and closeness of contacts between amino acids in the core are analyzed. The query sequence is evaluated for amino acid interactions that will correspond to those in the core and that will contribute the stability of the protein. The most energetically stable conformations of query sequence thereby provide predictions of the most likely three-dimensional structure. The inclusion of non-local interactions between amino acid pairs prohibits the use of the dynamic programming algorithm, because the assumption of independence in dynamic programming algorithms is no longer valid. Therefore, the contact potential approaches generally require more complicated algorithms to deal with the residue-residue contact term. They are more computationally expensive than the structural profile approach. Most existing programs employ heuristic approaches to solve the sequence-structure alignment problem. These approaches include double dynamic programming (Jones et al., 1992), frozen approximation (Godzik et al., 1992), Monte Carlo sampling algorithm (Bryant, 1996) and a divide-and-conquer algorithm (Xu and Xu, 2000). In the following, one of the best performing threading programs PROSPECT (Xu and Xu, 2000; Kim et al., 2003) is introduced. Some of the most recent well-designed knowledge-based energy potentials are given.

3.3.1 PROSPECT

PROSPECT (PROtein Structure Prediction and Evaluation Computer Toolkit) was developed by a research group at the Oak Ridge National Laboratory. It has two versions. The first version of PROSPECT (Xu and Xu, 2000) uses a divide-and-conquer algorithm to treat the pairwise potential strictly in aligning the target sequences to the templates. The divide-and-conquer algorithm solves the entire optimal alignment problem by recursively solving a series of sub-alignment problems

between sub-structures and sub-sequences, under various constraints, and then combining these sub-alignments in a consistent and optimal way. By using the divide-and conquer algorithm, PROSPECT could efficiently find a globally optimal threading alignment between the query and template proteins. Both pairwise contacts between spatially nearby residues and variable length alignment gaps are considered in the alignment algorithm. Four terms are included in the scoring function of the first version of PROSPECT. They are mutation potential, singleton energy potential, distance-independent pairwise contact potential and gap penalties. The singleton energy potential represents the structural environment fitness defined by secondary structure and solvent accessibility. The first version of PROSPECT performs very well in recognizing the fold recognition targets. However, it runs very slowly if the templates have complex interaction topologies and the target sequences are long.

PROSPECT-Π (Kim et al., 2003) discards the strict treatment of pairwise interactions to speed up the search for the optimal alignment in order to fulfill the genome-wise structure prediction. A two-stage threading strategy is developed. First, a query sequence is aligned to the templates by a dynamic programming algorithm regardless of pairwise contact potential. Both global and global-local alignment algorithms are employed. Then PROSPECT-Π calculates the distance-dependant pairwise score based on the existing alignment. The divide-and-conquer algorithm is used with all the energy terms including singleton and pairwise energies. Besides the non-pairwise z-score, the pairwise z-score is also calculated by randomly shuffling the sequence. A linearly combined z-score is calculated to select the best-fit templates. PROSPECT-Π runs very fast and greatly improves the alignment accuracy. Unfortunately, according to the CAFASP3 evaluation results (Fischer et al., 2003), PROSPECT-Π does not seem to work as well as the first version of PROSPECT in recognizing the fold recognition targets. The server of

PROSPECT is available at http://www.bioinformaticssolutions.com/products/prospect.php.

3.3.2 *Potential energy functions*

Besides the threading program PROSPECT (Xu and Xu, 2000; Kim et al., 2003), most current work is focused on the scoring functions only (Samudrala and Moult, 1998; Skolnick et al., 2000; Gatchell et al., 2000; Lu and Skolnick, 2001; Zhou and Zhou 2002; Lee and Duan, 2004).

Basically, there are two types of potential energy functions. The first class of potentials is called physical-based potential. They are built on the fundamental analysis of forces between atoms and they can be derived from the laws of physics. However, physical-based potentials have not been widely considered practical for protein threading due to the high-computation cost required for the calculation of free energy which should include an atomic description of the protein and surrounding solvent. To date, because of the continued improvement in computer speed, physical-based energy functions are showing signs of revival (Felts et al., 2002; Lee and Duan, 2004). The second class of potentials is called the knowledge-based potential. Compare to the physical-based potential, they are the mainstay in protein prediction programs. They extract information on the forces and energies from experimentally solved protein structures and measure the probability distribution of possible conformational arrangements of a protein sequence. Traditionally, they are also called "energy function" even if the scoring function does not reflect the real energy of proteins.

Knowledge-based potentials are derived from a statistical analysis of known protein structures. Normally, Bayesian method is used to deduce the knowledge-based potentials/energy function (Lathrop et al., 1998). Let

$P(T \mid S)$ be the probability of the query sequence S having in the same fold as the template T and $P(A \mid B)$ is the conditional probability of A given B. Let $A = A(1), A(2), \cdots, A(n)$ be an alignment between the query sequence and the template where residue in the sequence position i is aligned to the template position $A(i)$. Let $P(T \mid S, A)$ be the probability of the query sequence being aligned to the template according to the given alignment A. Then we have

$$P(T \mid S) = \max_{A} P(T \mid S, A) \tag{3.1}$$

According to probabilistic Bayesian theory, we have:

$$P(T \mid S, A) = \frac{P(T, S \mid A)}{P(S)} = \frac{P(S \mid T, A)P(T)}{P(S)} \tag{3.2}$$

If we assume that $P(T)$ is a uniform distribution, given a specific query sequence, then based on Equation 3.1 and 3.2, we have

$$P(T \mid S) \propto \max_{A} P(S \mid T, A) \tag{3.3}$$

Assuming the query sequence S is $a_1, a_2, \cdots, a_n$ and the template sequence is $T = t_1, t_2, \cdots, t_m$, $P(S \mid T, A)$ can be expanded as follows:

$$\begin{aligned} P(S \mid T, A) &= P(a_1, a_2, \cdots a_n \mid t_1, t_2, \cdots, t_m, A) \\ &= \prod_i P(a_i \mid t_{A(i)}) \prod_{i<j} \frac{P(a_i, a_j \mid t_{A(i)}, t_{A(j)})}{P(a_i \mid t_{A(i)})P(a_j \mid t_{A(j)})} \\ &\times \prod_{i<j<k} \frac{P(a_i, a_j, a_k \mid t_{A(i)}, t_{A(j)}, t_{A(k)})P(a_i \mid t_{A(i)})P(a_j \mid t_{A(j)})P(a_k \mid t_{A(k)})}{P(a_i, a_j \mid t_{A(i)}, t_{A(j)})P(a_i, a_k \mid t_{A(i)}, t_{A(k)})P(a_j, a_k \mid t_{A(j)}, t_{A(k)})} \cdots \end{aligned} \tag{3.4}$$

The first item of the right hand side of Equation 3.4 is the probability of one particular amino acid residue a_i being aligned at position $A(i)$ regardless of the alignment of other residues. The second item is the

probability of two residues a_i and a_j simultaneously being aligned to two specific template positions $A(i)$ and $A(j)$. The remaining items refer to the probability of the multiple sequence residues simultaneously occurring at multiple specific template positions. Since there is not enough experimental data to generate the other items apart from the first two in the right hand side of Equation 3.4, these items are often ignored. For computational convenience, $P(S \mid T, A)$ is often converted into its negative logarithm form, which is:

$$f(S \mid T, A) = -\log P(S \mid T, A) \tag{3.5}$$

The resulting f is called the energy function for protein threading. It is normally a sum of several items and often written as:

$$f(S \mid T, A) = \sum_i f_1(i, A(i)) + \sum_{i_1, i_2} f_2(i_1, i_2, A(i_1)A(i_2)) \tag{3.6}$$

In Equation 3.6, $f_1(i, A(i))$ is the singleton score when the amino acid in the sequence position i is placed to the template position $A(i)$. The singleton score could refer to the mutation term, secondary structure term and solvent accessibility. The mutation term is the probability of template residue at position $A(i)$ mutating to the sequence residue a_i. The secondary structure term refers to the probability of the sequence residue a_i occurring at the local secondary structure type. $f_2(i_1, i_2, A(i_1)A(i_2))$ represents the pairwise score when $A(i_1)$ and $A(i_2)$ are specially nearby and the residue in the sequence position i_1 is placed to the template position $A(i_1)$ while the residue in the sequence position i_2 is placed to the template position $A(i_2)$.

Depending on the different ways of generating singleton and pairwise scores, different knowledge-based potentials had been defined by

researchers. For example, GDV (Gatchell et al., 2000) is an atom-based free energy potential. It combines molecular mechanics with empirical solvation and entropic terms. KBP is a heavy atom distance-dependent knowledge-based pairwise potential developed by Lu and Skolnick (2001). KBP is designed with higher-resolution than those models using one or two points for each residue to represent a protein (Sun, 1993; Kolinski et al., 1998). Totally 167 different atom types are considered in KBP model. Zhou and Zhou (2002) developed an all-atom knowledge-based potential, DFIRE-A. A new reference state DFIRE (Distance-scaled, Finite Ideal-gas ReferencE) was established to construct the all-atom knowledge-based potential.

The performances of these knowledge-based potentials can be tested in two ways. They are the z-score from gapless threading and the ability to discriminate native structure from decoys.

3.3.5 TUNE

Different from those knowledge-based potential approaches, Lin et al. (2002) proposed a new approach on threading score. A BPNN is trained to predict the compatibility of amino acid residue side chain with its tertiary structure environments. A new scoring function is presented. The model is tested on benchmark problems of discrimination of native and decoy protein tertiary structures. It seems that the NN model is comparable with those pseudo-energy function approaches.

In their approach, each amino acid is described as main chain sphere and side chain sphere. The information entropy theory is used to get the optimal default radius of each side chain. The centre of the main chain sphere is placed on the carbonyl carbon. The residue contact is measured by the volume between amino acid side chain and its neighbours. A NN

model is trained on CATH (v2.0) database. The output of NN is transformed into log-odds score, which has the same formation as other threading scores based on potential energy. The model is called TUNE and two benchmark problems are used to evaluate the model.

In their paper (Lin et al., 2002), different NN models are considered and discussed with or without local structure and exposure description. A conclusion is given that both local structure and exposure characters of amino acid should be considered while making a threading program.

3.4 Research Framework

As stated above, when there are good templates in the protein structure library, good protein threading methods become very useful to reveal structural information for sequences. The vast amount of recent publications indicates the active research in the field of threading. However, the overall performance of current threading models is rather disappointing. For example, one of the better performing threading programs FUGUE (Shi et al., 2001), can only recognize 25% of homologous protein pairs with high confidence (99% specificity); GenTHREADER (Jones, 1999) can recognize correct fold with a low false-positive rate, but the alignment accuracy is comparatively low; PROSPECT (Xu et al., 2001) performed the best in the CASP4, but it runs very slow for long query sequences. Thus, more research work needs to be done to improve the performance of current threading model.

The structural profile method performs 3D-1D matching from structure templates. It is an established method for protein threading (Johnson et al., 1993; Rice and Eisenberg, 1997; Kelley et al., 2000; Shi et al., 2001). This research follows the structural profile approach to build a framework for

protein threading. The aim of this research is to develop a rapid, reliable, and automated protein threading model for a more comprehensive annotation of genomic sequences.

The literature review shows that the different threading approaches use at least one different threading component to improve the overall threading performance, such as, the representative of the protein, the scoring function, the alignment algorithm and the way alignment significance is assessed. Within the structural profile approach, Bowie et al. (1991) described the structure environment in term of solvent accessibility, contact with polar protein atoms and secondary structure type. Rice and Eisenberg (1997) defined structure position by one of seven residue classes, three secondary structure classes and two burial classes. In FUGUE (Shi et al., 2001), the structure environments are defined in three groups, which are main-chain conformation and secondary structure, solvent accessibility and hydrogen bonding status. They demonstrated that by including structural information, the performance of fold recognition could be improved. However, the features selected as classes or groups cannot precisely describe all the complex 3D structures. These coarse-grained descriptions can be refined by a 3D-1D mapping. Inspired by TUNE model (Lin et al., 2002) in which a NN is used at the amino acid residue level to map residue-structure compatibility, an idea of generating environment-specific amino acid substitution probabilities (3D-1D mapping) by NNs is proposed. More precise structural information can be extracted by NNs. The performance of the threading model is therefore expected to be improved.

Unlike those atom-based threading models (for example, DASEY, Mallick et al., 2002), in which the threading computation is expensive, the framework for protein threading is proposed on residue level. A fast threading model is expected.

The research work is outlined in Figure 3.2. Basically the research work can be divided into two parts. The left part is the main research focus. To improve the performance of current threading models, a framework for automated protein threading (MESSM; **M**ixed **E**nvironment-**S**pecific **S**ubstitution **M**apping) is designed with NNs and SVMs. The main research is extended to design a threading score (TES; **T**hreading with **E**nvironment-specific **S**core) following contact potential approach, which is the right part of Figure 3.2.

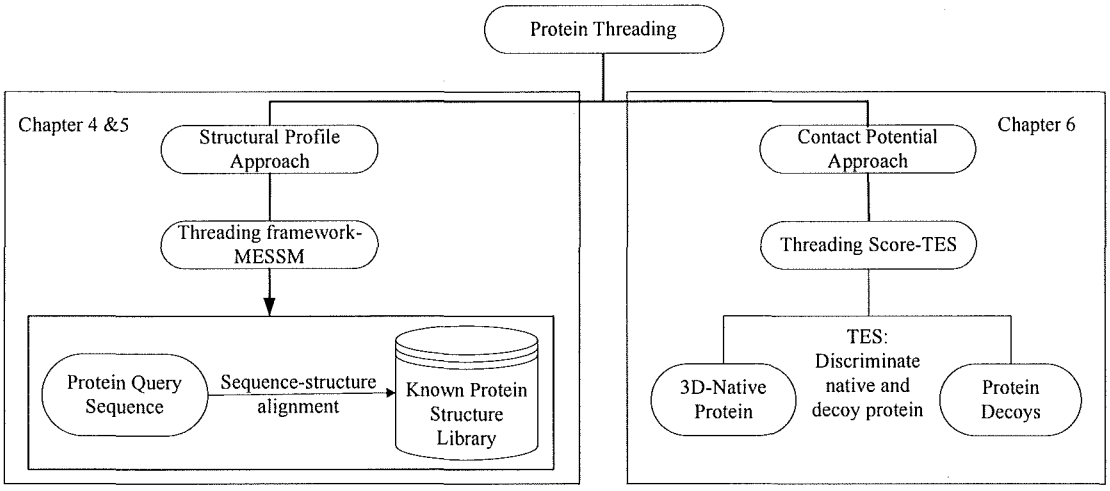


Figure 3.2 Schematic diagram of this research

In this thesis, chapters four and five report the main research work, which is to design and evaluate a framework for protein threading following structural profile approach. The designed threading score following contact potential approach will be reported in Chapter 6. A brief description of the two parts of the research work is given:

A framework for protein threading by structural profile approach (MESSM).

For developing and evaluation a framework for protein threading, the following steps will be adopted.

1) Unlike previous work (Fischer and Eisenberg, 1997; Shi et al., 2001) in which the structural environments are defined as classes or groups, NNs will be adopted in this research to generate the amino acid substitution probabilities at any structural environment. NNs have been shown to be an efficient tool in solving several kinds of problems in bioinformatics (reviewed in Chapter 2, Section 2.3.3). Lin et al. (2002) successfully applied NNs at residue level to predict residue-structure compatibility. Thus, choosing a NN can be a right choice to generate the environment-specific substitution probabilities in this research.

   In the first step, given a residue with its structure environment, the probability that it can be replaced by other residues will be obtained by NNs. A substitution mapping for protein threading will be constructed by log-odds score from the output of NNs. A representative fold library will be built in the formation of log-odds score matrixes. To do this, sequences with representative fold are selected from SCOP first; then amino acids of each selected protein sequences are encoded according to their environment description; and finally, a NN model is used to give the output as a profile for each sequences. Each profile in the library is a matrix with size of (protein length) $\times 20$. Each line of the matrix represents an amino acid with its environment, the probabilities it can be substituted by 20 kinds of amino acids.

2) Previous research (3D-PSSM, Kelley et al., 2000; SPARKS, Zhou and Zhou, 2004) demonstrated that by including more information of known proteins, the threading performance can be improved. The environment-specific substitution mapping generated from NN only includes protein structural information. Amino acid substitution matrices (such as, PAM & BLOSUM) are built from sequences database with useful evolutionary information of protein

sequences. Following consensus theory, a mixed substitution mapping can be created by linearly combining the two parts.

In step two, a mixed environment-specific substitution mapping (MESSM) will be built with an optimized parameter. Dynamic programming (Needleman and Wunsch, 1970; Smith and Waterman, 1981) will be employed to align the probe sequence with structures in the fold library.

3) GenTHREADER (Jones, 1999) and WURST (Torda et al., 2004) both use NNs instead of a z-score or P-value to evaluate the sequence-structure alignment. In their model, the template selection is treated as a classification problem. Their experimental results showed that both models can automatically predict the correct fold with a comparatively low false-positive rate. Since SVMs are a new binary classification method and have been demonstrated to have superior performance in various problems compared to NNs (Ding and Dubchak, 2001), in this research, each alignment will be evaluated by SVMs.

   In step three, a SVM will be trained to evaluate the significance of sequence-structure alignment. The one with the highest score will be chosen as the best template for the query sequence.

4) With the combination of the three steps introduced above, a new framework for protein threading will be built. To evaluate the effectiveness of the proposed framework, benchmarks will be used. Both the fold recognition performance and alignment accuracy will be verified. The results will be compared with current threading models.

Threading score following contact potential approach (TES).

Although the structural profile approach is the established method for protein threading, the most successful protein threading method is based on contact potential techniques (Xu and Xu, 2000; Kim et al., 2003). Thus, this research will be extended to build a threading score following contact potential approach.

Since residue contact calculation is the most important factor in protein prediction, a good calculation of residue contacts would play a fundamental role in protein threading models. In this research, using the new residue contact measuring scheme developed in the MESSM model, the compatibility of a residue in sequence with its structural environment will be presented by NNs. The probabilities from the NN output will be transformed into a log-odds score that can determine which residue best fits its environment. The effectiveness of the score will be tested on benchmarks to discriminate protein native and decoy sets. The results will be compared with other threading scores based on energy potentials. Also, the results will be compared with the NN based TUNE model (Lin et al., 2002), which use a different residue contact measuring scheme.

With the proposed research framework, the new threading method is expected to be effective and efficiency. That means, both the threading framework (MESSM) and the threading score (TES) should have a comparable performance with current research work if not better. Also, the MESSM should be a rapid, automated threading method to match the requirement of fast genome sequencing in the post-genome era.

CHAPTER 4 THREADING (STRUCTURAL PROFILE APPROACH) USING NEURAL NETWORKS AND SUPPORT VECTOR MACHINES

As reviewed in Chapter 3, threading techniques could broadly be divided into two categories: one performing 3D-1D matching using evolutionary relationship, which is normally called the structural profile method; the other using pairwise interaction potentials, which is called contact potential method. Since the former handles the proteins in family and superfamily level, it is also called "homology recognition" (Williams et al., 2001). Profile (Bowie et al., 1991) and Hidden Markov model techniques (Eddy, 1998) commonly fall into this category. In the homology recognition method, a sequence can be aligned to known protein folds using energy functions or probabilistic scoring schemes (e.g. Bowie et al., 1991; Rice and Eisenberg, 1997; Jones, 1999; Shi et al., 2001).

This chapter proposes to design a new framework for protein threading following the structural profile approach. Thus, the protein threading problem could be considered in the following version:

Step one, given a protein sequence called the target (query) and a protein structure called the template, it is required to look for a suitable alignment of the target sequence onto the template structure. Therefore, a structural profile of a template should be built first. Then a sequence-profile alignment should be implemented. Finally, a score function will be needed to be given for the alignment.

Step two, given a protein target sequence and a representative fold database, a list of sequence-template alignments with scores is obtained. The best template is chosen based on the alignments' score. In order to do this, a representative fold library need to be built first. Then an evaluation method is used to choose the best template for the target.

In section 4.1, an overview of the proposed framework for protein threading is given. The outline and the key features of the framework are introduced. The details of the threading framework design are described in sections 4.2 to section 4.6. A summary is given in section 4.7.

4.1 Overview

In this research, a new framework of automated protein threading with **M**ixed **E**nvironment-**S**pecific **S**ubstitution **M**apping, MESSM, is proposed using NNs and SVMs (as shown in Figure 1.1). The proposed framework has three key features consisting of three main parts. They are: building the fold profile library, mixed substitution mapping and confidence evaluation, as outlined in Figure 4.1, Figure 4.2 and Figure 4.3 separately.

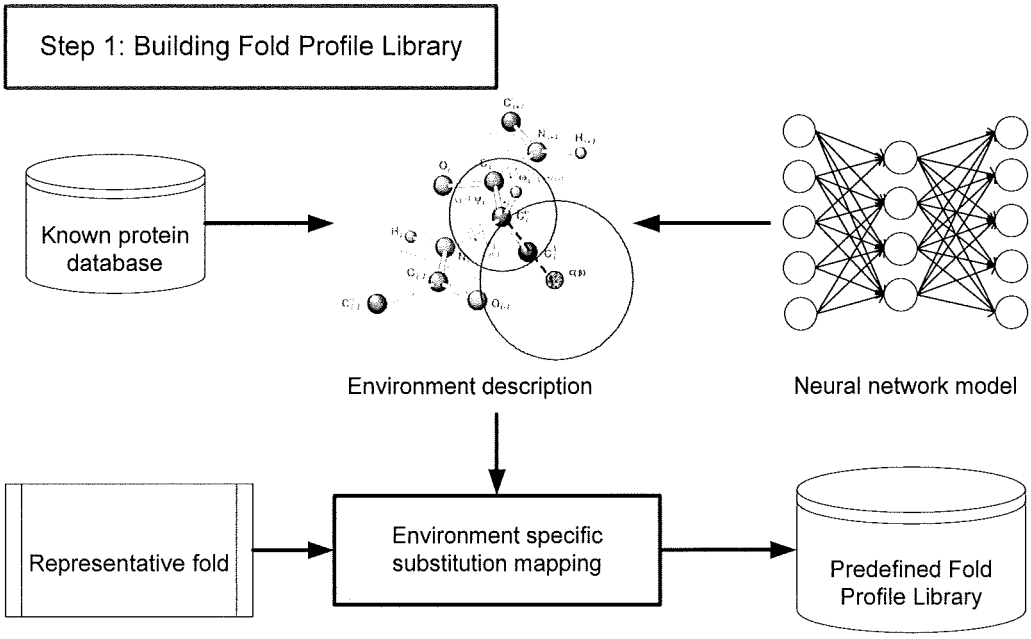


Figure 4.1 Step one of MESSM, building the fold profile library.

1) *Building the fold profile library.* Given a known protein structure database, the structural information of a protein is described by each amino acid with its environment description. Unlike the environment-specific amino acid substitution tables in which the structural environments are defined as groups, NNs are trained to extract more precise structural information with amino acid residue-level environmental description. The substitution probability of each pair of amino acids at any chosen structural environment can be generated from the trained NN and transformed into log-odds scores. A predefined representative fold library is built as profiles on the substitution probabilities. The details will be discussed in Section 4.2 to Section 4.4.

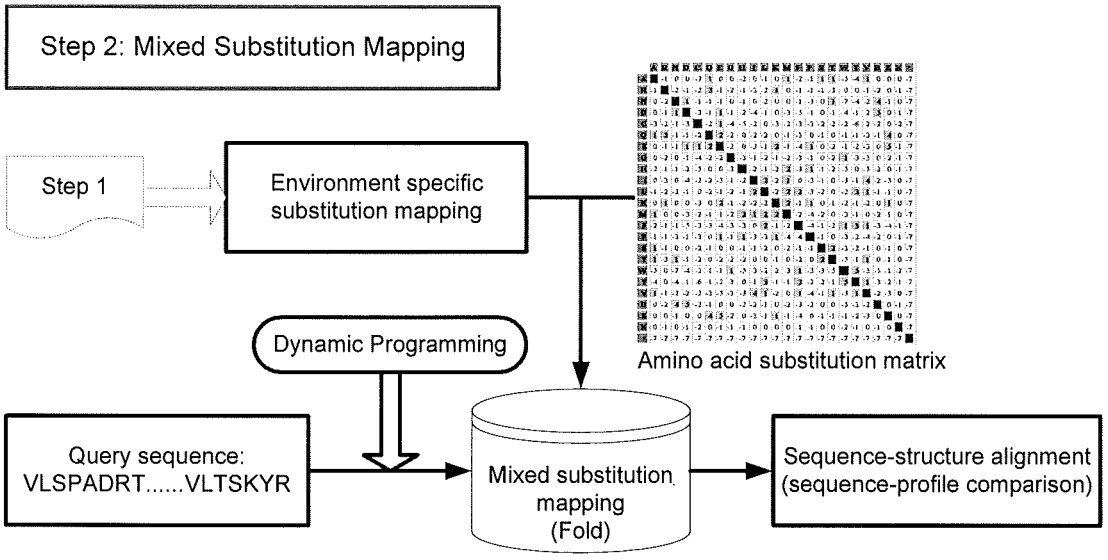


Figure 4.2 Step two of MESSM, mixed substitution mapping.

2) *Mixed substitution mapping*. According to consensus theory, linearly combine information from both the structurally-derived substitution score (obtained from the Step 1 of the MESSM) and the sequence profile from well-developed amino acid substitution matrices (for example, BLOSUM30) to produce a mixed substitution mapping. Thus, given a query sequence, the sequence structure alignment could be acquired by dynamic programming with the mixed substitution mapping. The details will be discussed in Section 4.5.

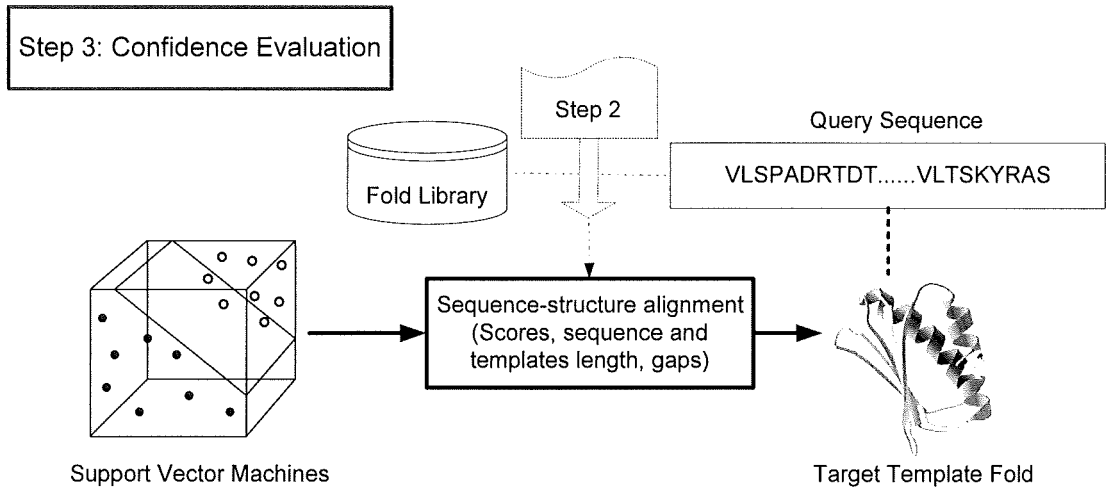


Figure 4.3 Step three of MESSM, confidence evaluation.

3) Confidence evaluation. A SVM is employed to measure the alignment significance between the protein query sequence and fold profile (obtained from Step 2 of the MESSM). The best template is chosen for the query sequence. The details will be discussed in Section 4.6.

4.2 Protein Environmental Description and Residue Contact

In this research, our framework is proposed using the residue level structural environment description.

4.2.1 Description of Structural Environments

According to the work of Bryant and Lawrence (1993) in developing contact potentials for protein threading, each amino acid residue is considered to be composed of a side chain fragment, which is different for each of the 20 amino acids, and a main chain fragment that is the same for all the amino acids. Two amino acids interact when their side chains are in

contact or the side chain of one amino acid is in contact with the main chain of the other's.

In this research, each amino acid residue is also described using two spheres: the sphere of main chain and the sphere of the side chain (Figure 4.4).

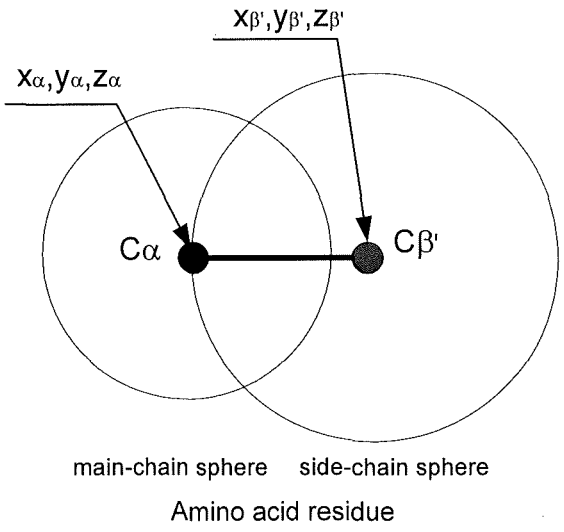


Figure 4.4 Description of structure environment. Each residue is represented by a main chain sphere with centre C_α and a pseudo side-chain sphere with centre $C_{\beta'}$.

The calculation of the side chain radius follows Lin's approach (Lin, et al., 2002): All spheres are considered to have the same density, so the radius of each is proportional to the cube roots of its mass. The main chain mass is 56.0D. The radius of an Alanine side-chain sphere is 1.7Å and its mass is 15.1D, so the radius of other amino acid residue can be computed. The side chain radius of all the amino acids are listed in Appendix I. The pseudo side-chain centre is built by extending the bond between the alpha and beta carbon to the radius of the side-chain.

As shown in Figure 4.4, two pairs of coordinates are used to determine the two spheres and describe the structural environment of each residue. They are:

1) The coordinates of backbone alpha carbon are used to determine the main chain sphere as most of other research work does (Bryant and Lawrence, 1993). The coordinate values are extracted from the high-resolution structure file in PDB ($x_\alpha, y_\alpha, z_\alpha$; Bernstein et al., 1977; Berman, et al., 2000).

2) The coordinates of pseudo side-chain centres ($x_{\beta'}, y_{\beta'}, z_{\beta'}$; Lin, et al., 2002) are used to determine the side-chain sphere.

4.2.2 Residue contact measurement

In general, it has been agreed that contacts among residues constrain protein folding and characterize different protein structures. Therefore, the residue contact calculation is the most important factor in protein prediction, especially for those interactions between residues that are distant in the sequence (long-range interactions). The basic assumption is that the conformation of protein structure follows the Boltzmann distribution: the probability of observing the contact is proportional to log energy states, and native protein structures should have lowest energy states. Provided that residue contacts are known for a protein sequence, the major features of its 3D structure could be deduced by applying reconstruction method (Bohr *et al.*, 1993). A good calculation of residue contacts would play a fundamental role in protein threading models (Sippl, 1990; Huang *et al.*, 1995; Lathrop and Smith, 1996; Taylor, 1997). With less contact environmental description, the model may miss important information and lead to the wrong solution. With too much contact description, the model may import some noise, which may reduce the efficiency of the model.

In some early work, the two amino acids' contact is calculated by a simple distance cutoff (for example, 10Å by Jones et al., 1995). This is improved

by importing pseudo-side chain position (Taylor, 1997). Further development in this area leads to a specific research topic of residue contact map prediction (Lund et al, 1997; Olmea and Valencia, 1997; Fariselli and Casadio, 1999; Pollastri and Baldi, 2002). In Lin's (2002) TUNE model, the contact is measured by the overlapping volume between side chain and its neighbours.

In this research, a new residue contact measurement is proposed. It is built to reflect the fact that if the space between two amino acids is larger than one water molecule or a third residue, then it means they are too far to have contact. Thus, for each residue under consideration, other residues in the protein sequence are regarded as its neighbours and are considered to have a contact when either one of the following two conditions are true:

1) Side chain -side chain contact: the distance between two side-chain centres is less than the sum of radius of both side-chains plus twice the radius of the solvent molecule (Figure 4.5);

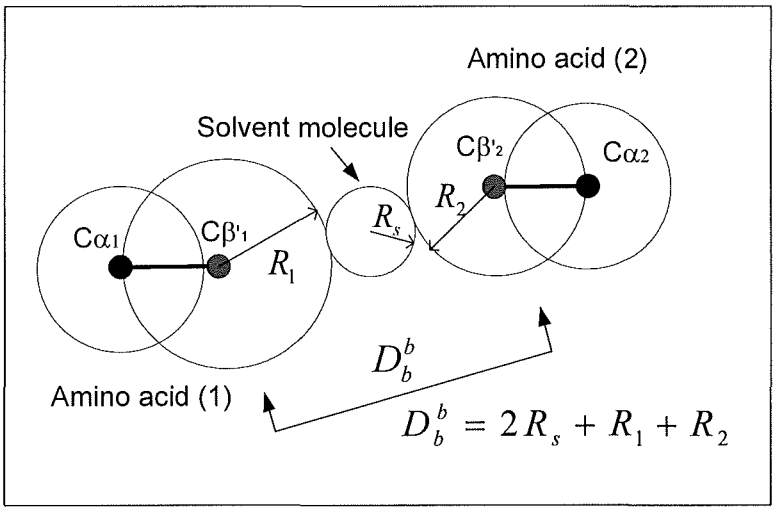


Figure 4.5 Side chain to Side chain contact: the distance between two pseudo side-chain centres of two amino acid residues is less than $D_b^b = 2R_s + R_1 + R_2$, which is the sum of side-chain radius of two amino acid residues (1 and 2) plus twice the radius of the solvent molecule (R_s).

2) Side chain—main chain contact: the distance between side-chain centre of one residue and the main-chain centre of others is less than the sum of radius of one side-chain and one main-chain plus twice the radius of the solvent molecule (Figure 4.6).

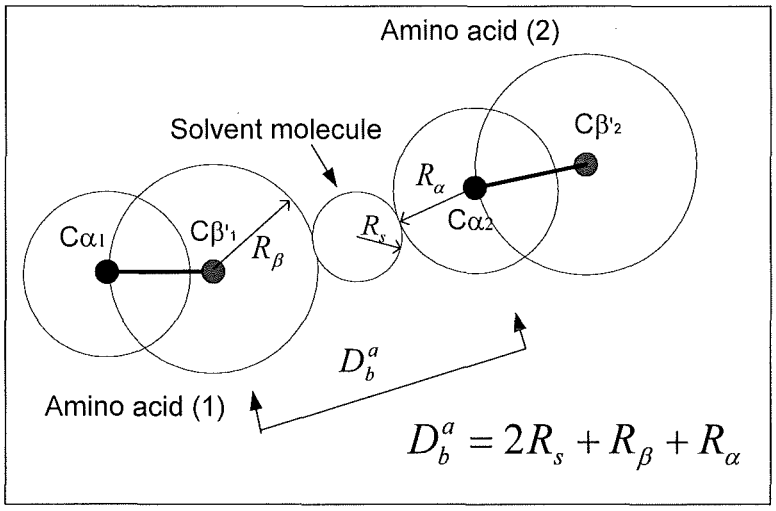


Figure 4.6 Side chain to main chain contact: the distance between pseudo side-chain centre of one amino acid residue and the backbone centre of the other amino acid residue is less than $D_b^a = 2R_s + R_\beta + R_\alpha$, which is the sum of side-chain radius of one amino acid (1) plus the main chain radius of amino acid (2) and twice the radius (R_s) of the solvent molecule.

The solvent radius is set to 1.4Å (the radius of water molecule) in this research.

4.3 The Artificial Neural Network model for environment-specific substitution

As introduced in Chapter 2, ANNs are a new generation of information processing systems that are deliberately constructed to make use of some of the organization principles that characterize the human brain. They are parallel computational models comprised of densely interconnected

adaptive processing units. It has been shown that NNs are more efficient tools in solving several kinds of problems than other approaches (Baldi and Brunak, 2001). For example, NNs are shown to be the first protein secondary structure prediction method to surpass a level of 70% overall three-state accuracy (Rost et al., 1994). They have also significantly improved the accuracy of structural classes prediction (Chandonia and Karplus, 1995).

ANNs are very well suited for domains with an abundance of data and lack of clear theory, which is precisely the case in the protein threading problem. Thus, in this research, a three-layered fully connected BPNN with 45 input neurons, 20 output neurons and 30 hidden neurons is used to predict an amino acid residue with its environmental description, and the probabilities that it could be replaced by other amino acid types, as shown in Figure 4.7. The reason for choosing the BPNNs in this research is that the BPNNs are currently the most general-purpose and commonly used NN paradigms, which achieve their generality because of the gradient-descent technique used to train the networks.

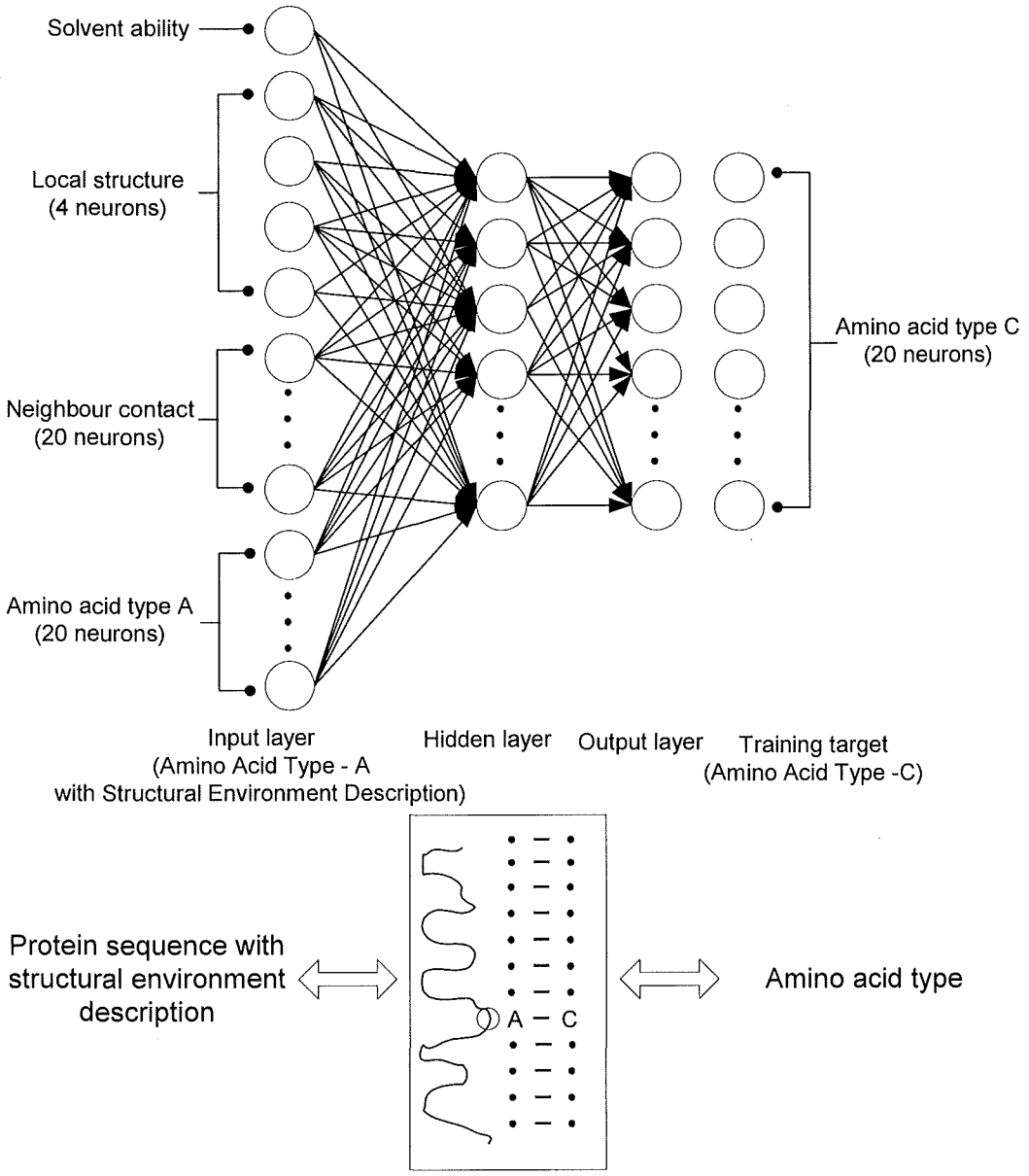


Figure 4.7 The NN model for training. Three layered feedforward NN: 45 input neurons describing amino acid with its environmental structure, 30 hidden neurons and 20 output neurons.

4.3.1 Input representation

Unlike most of the NN approaches to protein fold recognition (Ding and Dubchak, 2001; Baldi and Brunak, 2001), whose input of NN represents a whole protein sequence, the input of the NN in MESSM is an amino acid.

Since on average each protein has 200~300 amino acids, our training data is 200~300 times more than those methods based on whole protein sequence.

In total, 45 input units are used to describe the features of amino acid residue with its structural environment. Given an amino acid a_i on protein sequence $S = a_1 a_2 \cdots a_n$ with known structure, C_{ij} represents the other amino acid a_j has contact with a_i. One input unit is used for residue solvent ability, measured by the sum of all the residue contacts, which is $\sum_j C_{ij}$. Four units are used to represent a local structure, which is calculated by the distances from the alpha carbon to the alpha carbons of amino acid pairs of (a_{i-4}, a_i), (a_{i-2}, a_i), (a_{i+2}, a_i) and (a_{i+4}, a_i). Twenty units are represented the twenty amino acids of a_i, which is encoded by orthogonal encoding scheme. The left twenty input units are employed to encode the neighbour contacts of a_i. For each neighbour a_j, a value of "one" is added to the corresponding unit according to its amino acid type. A value w_{ij} is computed as the weight of "one" to reflect neighbour contact C_{ij}, so the closer the neighbours are the greater the contact influence they have.

$$w_{ij} = \begin{cases} D_b^b - d_{ij}^b \\ D_b^a - d_{ij}^a \end{cases} \tag{4.1}$$

where D_b^b, D_b^a are distance thresholds according to two kinds of neighbour contacts described in section 4.2.2 (Figure 4.5 and Figure 4.6), d_{ij}^b and d_{ij}^a are the distances between the two amino acids (a_i, a_j) side-chain centres and one side-chain and one main-chain centres separately.

4.3.2 Methodology for Neural Network Training

The methodology used in the training of NN is summarized as follows:

1) As one hidden layer with sufficient neurons can map a training set with arbitrary precision (Cybenko, 1989), the proposed ANN model is focused on one hidden layer, one input layer and one output layer. As shown in Figure 4.7, a three-layered feed-forward NN with 45 inputs and 20 outputs was used in this study. The input of 45 real numbers describes the amino acid type with its structural environment, as described above. The target of NN is the amino acid type from the results of structural alignment, which is encoded by the orthogonal encoding scheme (20 units). Various network architectures were tested by changing the number of neurons in hidden layer from 10 to 40. The 30-hidden-neuron model was selected due to its best performance.

2) In order to ensure that the solution is reasonably close to the global minimum, the NN is trained with 10~20 different starting conditions, that is, random initial weights and biases.

3) The standard logistic sigmoid activation function is used for the hidden layer and the softmax activation function for the output layer due to the output range (0 to 1). The relative entropy error is used to measure the performance of NNs (Baldi and Brunak, 2001).

4) The training algorithm employed here is the back-propagation algorithm to minimize the mean difference between the predictions and real amino acid types.

5) The training is stopped using an eight-fold cross-validation approach. In eight fold cross-validation, the data is divided into

eight subsets of approximately equal size. The data is partitioned into training and test data in eight different ways. After training the NN with a collection of seven subsets, the performance of the NN is tested against the eighth subset. This process is repeated eight times so that every subset is once used as the test data.

6) Each input value is scaled to lie in the range of 0~1 using standard logistic function:

$$Input = \frac{1}{1 + e^{-a(x-b)}} \tag{4.2}$$

where x is the raw input value and a and b are constants. In this work, a=1 and b=10 (McGuffin and Jones, 2003).

4.3.3 Datasets

As mentioned in Section 3.2.4 above, several classifications of protein architectures are publicly available including SCOP (Murzin et al., 1995; Lo Conte, L., et al., 2002), CATH (Orengo, et al., 1997) and FSSP/ DALI (Holm and Sander, 1997). SCOP is manually constructed by Dr. Alexei Murzin, thus it is independent of any specific sequence or structure comparison algorithm. Both CATH and FSSP, on the other hand, are built more or less automatically from structural alignments. While the CATH and FSSP classifications use protein chains as the object of interest, SCOP breaks proteins into domains as a result of eliminating the problem of placing multi-domain proteins in the classification hierarchy. The reason for choosing SCOP as the training and testing data, instead of using CATH and FSSP, is due to the high quality of the database and the use of domains instead of complete protein chains.

Therefore, the structure classification database of SCOP (v1.65) (Lo Conte et al., 2002) is used to select the training and testing data sets for NNs.

Since the aim of this research is to discover the relationship of proteins with long distance evolution, only those proteins with lower than 40% sequences similarities are considered. Thus, 1150 pairs of non-redundant domains are selected. 881 pairs are in the family level, 269 pairs in the superfamily level.

All the protein pairs are aligned using structural alignment program-FLASH (**F**ast a**L**ignment **A**lgorithm for finding **S**tructural **H**omology of proteins; Shih and Hwang, 2003). An example of protein pair's alignment using FLASH could be found in Appendix Π. Totally 190,603 residue pairs are used to train NNs.

4.3.4 Neural network training result

The BPNN is trained by using various network architectures with the number of neurons in hidden layer from 10 to 40. Each architecture is trained with 10~20 different starting conditions. The average training and test error for the different architecture is shown in Table 4.1. The best performance NN is the one with 30 hidden neurons. Figure 4.8 shows the curve of its training error.

Hidden neuron	Average training error	Average test error
10	2.35846	2.36202
12	2.35318	2.35981
14	2.34086	2.35284
16	2.32908	2.33862
18	2.33466	2.34070
20	2.32209	2.33098
22	2.32111	2.32841
24	2.32002	2.32651
26	2.31332	2.31970
28	2.31076	2.31819
30	**2.30772**	**2.30982**
32	2.30905	2.31124
34	2.31254	2.32268
36	2.31967	2.32563
38	2.32721	2.33016
40	2.32814	2.33569

Table 4.1 The training and test error for the different ANN architectures

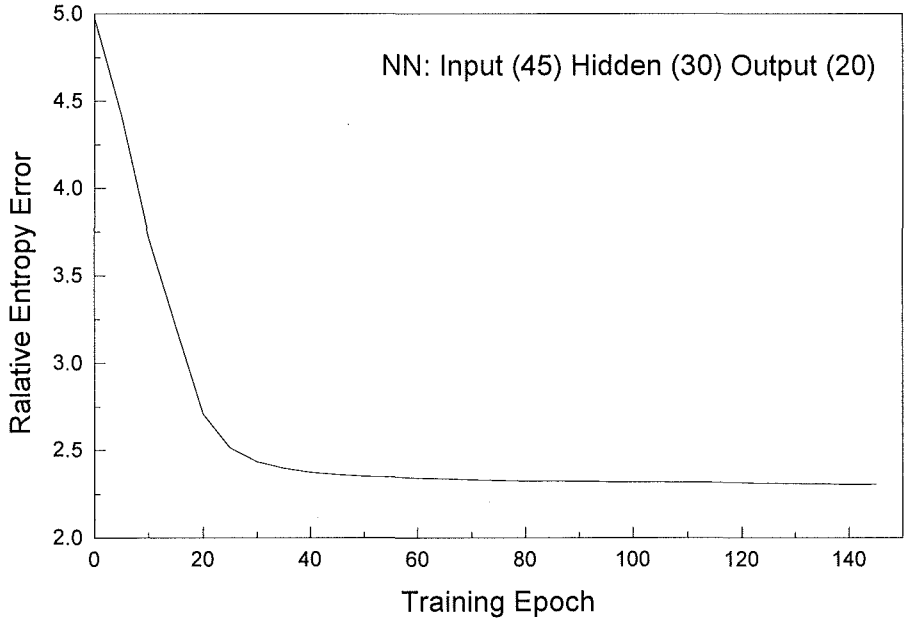


Figure 4.8 Relative entropy errors of the training. The training stopped at 145 epochs and the error is 2.304 (the best performance NN).

4.3.5 *Substitution scores*

Let $P(x \mid y, E)$ be the frequency of observing residue y in environment E replaced by residue x. Given a residue y in a protein structure E, its type and structure environment are encoded as input of the trained NN model. The output of the NN is the predicted substitution probability $P(x \mid y, E)$. A log-odds score of the substitution is given by:

$$S(y, E \rightarrow x) = \ln(\frac{P(x \mid y, E)}{P(x)}) \tag{4.3}$$

where $P(x)$ is the occurrence of the residue x in the sequence. The higher the logarithm likelihood score is, the better y residue is replaced by x in the structure environment E.

4. 4 Representative fold profile library

To do protein sequence-structure alignment, the additional structure information from protein structure templates should be included in order to detect remote evolutionary relationships, which could not be detected by sequence alignment program. A popular method is to generate a Position-Specific Scoring Matrix (PSSM), also called one-dimensional (1D) profile, from protein structure templates (Brenner et al., 1998; Lackner et al., 1999).

Different methods have been developed to generate PSSMs (Henikoff and Henifoff, 1994) using multiple alignment, predicted secondary structures and other features (e.g. Henikoff and Henifoff, 1997; Elofsson, et al., 1996; Rost, et al., 1997; Zhang and Eisenberg, 1994). Bowie et al. (1991) first proposed the threading method using a 1D profile. They calculated amino

acid preferences for structural environments defined in terms of solvent accessibility, contact with polar protein atoms and secondary structure type. Based on these preferences, one-dimensional profiles were generated from protein structure and used to align to sequence by dynamic programming algorithm.

In the 3D-1D substitution matrix approach by Rice and Eisenberg (1997), each structure position was defined by one of seven residue classes, three secondary structure classes and two burial classes. Each sequence position was defined by one of seven residues classes and three predicted secondary structure classes. The matrix scores the substitution between residues of different classes. A dynamic programming algorithm is used with these scores to align a probe sequence with representative structures in the fold library after the prediction of probe secondary structure (Rost, et al., 1997). In their program, information from multiple sequence alignment of probe sequence is used to predict secondary structure and residue exposure. Recently, multiple alignments of probe sequence and target structure are used for building of 1D profiles (e.g. Kelley, et al., 2000; Shi, et al., 2001).

In this research, the PSSM is generated in a different way from previous work in which the structural environments are defined as groups. A BPNN is trained to extract more precisely structural information with a protein residue-level environmental description. With the additional structural information from protein 3D templates, predicted residue substitution probabilities are expected to be improved. All the template protein structures could be transformed into 1D profile.

The representative fold library is built on the basis of 3D-PSSM (Kelley et al., 2000) but keeps only SCOP (Lo Conte et al., 2002) sequences. Also the proteins with low resolution (lower than 4Å) are not included. So, in total 4775 protein templates are selected as representative folds for our fold

library. The fold library in 3D-PSSM is an up-to-date fold library with SCOP-1.53. It has a good coverage on current available folds. By filtering out the low resolution proteins and the personal designed folds by 3D-PSSM, the representative fold library in this research is good enough for the experimental evaluation. For each sequence $S = a_1 a_2 \cdots a_n$ in the fold library of length n, where a_i is one of the 20 amino acids, a_i and its structural environment are encoded as input of a trained NN. The probabilities of a_i replaced by each of the 20 types of amino acids are generated from the outputs of NN and the values are transformed into log-odds scores as described in equation 4.3. A matrix of $n \times 20$ (1D profile) is built for each fold in the library. An example is shown in Figure 4.9.

	20 Type of Amino Acid												
	A	R	N	D	C	Q	..	..	S	T	W	Y	V
F(E)	0.12	0.28	-0.30	0.75	1.16	-0.10	:	:	0.08	-0.11	0.89	-0.05	-0.97
E(E)	-0.26	-0.21	-0.08	-0.01	0.96	0.07	:	:	0.01	0.08	1.06	0.15	-0.15
N(E)	-0.33	0.42	0.16	0.23	1.03	0.25	:	:	-0.73	0.44	1.04	0.03	-0.81
:	:	:	:	:	:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:	:	:	:	:	:
A(E)	-2.08	-0.84	-3.23	0.19	-0.27	-1.80	:	:	-0.24	1.04	2.05	1.44	0.64
V(E)	-0.24	-0.16	-0.06	0.10	0.90	0.13	:	:	-0.05	-0.04	1.07	0.57	-0.01
K(E)	-2.26	-2.03	-0.03	-1.17	-1.25	-1.72	:	:	-4.59	-3.42	-0.25	-0.83	1.81
sequence with structure													

Figure 4.9 An example of predefined fold profile

4.5 Mixed Substitution Mapping

4.5.1 Substitution Scores

Amino acid substitution matrices are built from sequences database with useful evolutionary information of sequences and provide the foundation for many search techniques. In the 1D profile generated above by NNs, the environment-specific substitution mapping import structural information into the residue alignment. Follow consensus theory, a mixed substitution mapping is proposed to combine the environment-specific mapping and the BLOSUM30 amino acid substitution matrix. Thus, for each alignment, if these two measurements agree, then positive consensus create a good alignment; if one gives a strong objection, the alignment is in doubt even if the second one shows a positive signal.

Suppose $S(x \mid y)$ represents the amino acid substitution matrix BLOSUM30, and the environment-specific substitution mapping given from the output of NNs is $S(y, E \rightarrow x)$. The combined substitution mapping $M(x \mid y, E)$ is defined as a linear combination of $S(y, E \rightarrow x)$ and $S(x \mid y)$:

$$M(x \mid y, E) = \mu S(y, E \rightarrow x) + (1 - \mu) S(x \mid y) \tag{4.3}$$

The parameter μ is a constant between zero and one and is optimized by Fischer's benchmark (descried Chapter 5) in this research.

4.5.2 Dynamic Programming

With the 3D-1D substitution mapping described above, the template 3D protein structures could be transformed into 1D profiles with mixed

substitution scores. So, the alignment of a sequence $S = a_1 a_2 \cdots a_n$ of length n, where a_i is one of the 20 amino acids, into a structure $X = x_1 x_2 \cdots x_m$ with m residues, where x_j is the 1D profile, could be considered as sequence alignment. To obtain the optimal alignment between two sequences, dynamic programming (Needleman and Wunsch, 1970; Smith and Waterman, 1981) can be used (as introduced in Section 2.3.2). Dynamic programming is a good method for finding an optimal alignment when the substitution score can be obtained at each position of the alignment. Many fold recognition methods use the dynamic programming algorithm in various forms, including local alignment (Jones *et al.*, 1992), global alignment (Bowie *et al.*, 1990) and the global-local alignment (Fischer and Eisenberg, 1996; Rice and Eisenberg, 1997). In this research, a probe sequence is aligned to 1D profiles using a global-local dynamic programming algorithm (Fischer and Eisenberg, 1996). This algorithm is shown to have better performance than both global dynamic programming (Needleman and Wunsch, 1970) and local dynamic programming (Smith and Waterman, 1981).

The global-local alignment algorithm does not penalize unmatched N- or C- termini segments in the probe sequence (as in the local alignment), but it does penalize any gaps in the target structure (as in the global alignment with ends penalization). Thus, as a result, the global-local alignment has two features:

1) The possibility of obtaining higher scores for relatively short, local matches is reduced since all the positions in the structure are counted in the alignment;

2) If the fold is larger than the probe sequence, more gaps need to be included, and the score of this match would be low. Therefore, the

tendency of obtaining higher scores for large structures is also reduced.

The constructed fold profile library contains only SCOP (Lo Conte et al., 2002) sequences whose structures are largely composed of domains, whereas the query sequence may contain more than one domain. In other words, folds are generally shorter than probes in this research. That is why the global-local alignment algorithm is chosen. The entire sequence of a library entry should be aligned within the query sequence.

4.5.3 Gap Penalty

In general, the gap penalty could be a constant or a function. In this research, the affine gap penalty is used. The score for a gap of length x can be presented as:

$$GAP = g + g'x \tag{4.4}$$

where g is the cost of opening a gap and g' is the cost of extending a gap. The parameters g and g' could be optimized by alignment evaluation in the following section (Section 4.6.3).

4.6 Confidence evaluation

4.6.1 Overview

In a fold recognition program, a target sequence is aligned with all structures in a fold library using dynamic programming. However, how to choose the best template based on alignments is also critical to the success of protein threading. The sequence-template alignment score cannot be

directly used to rank the templates due to the bias introduced by the residue composition and the number of alternative sequence-template alignment. To evaluate the best fit templates, an early statistical method is to use z-score (Flockner et al., 1995). However it has been shown that the z-score is not effective (Marchler-Bauer and Bryant, 1997). So, statistical P-value (Karlin et al., 1990) have been used to do the task. A P-value estimates the probability of having alignment scores between two random sequences higher than a particular value, and has been successfully applied to sequence alignment (Xu, et al., 2002). Recently, NNs have been used to evaluate alignment (Jones, 1999; Xu et al., 2002; McGuffin and Jones, 2003). The neural-network based assessment capability has been implemented in CASP4 and gained success.

The NN method treats the template selection problem as a classification problem. It required no human intervention in the prediction process. The automated method makes it possible to analyze many thousands of genomic sequences. In GenTHREADER model (Jones, 1999), a NN model is trained with the length of two protein domains, alignment length, the alignment score, and the scores of sequence-structure compatibility from pseudo energy function to predict the significance of the alignment.

SVMs are a new binary classification method developed by Vapnik and coworkers (Vapnik, 1995; Burges, 1998) and successively extended by a number of other researchers (Osuna et al., 1997; Joachims, 1999). During the past few years, the SVM has been broadly applied in the area of bioinformatics based on two main motivations (Noble, 2004). First, many biological problems involve high-dimensional, noisy data, and the difficulty of a learning problem increases exponentially with dimension. It has been a common practice to use dimensionality reduction to resolve these problems. The SVM can cope with high dimensional problems by maximizing the margin, which is characterized by the distance between the nearest training point and the optimal separating hyperplane.

Empirically, it has been shown to work in high dimensional spaces with remarkable performance (Cristianini and Shawe-Taylor, 2000). Second, in contrast to most machine learning methods, SVM can easily handle non-vector inputs, such as variable length sequences or graphs. These types of data are common in biology applications.

Since SVMs have been demonstrated to have superior performance in various problems compared to NNs (Ding and Dubchak, 2001), in this research, a SVM is employed to evaluate the sequence-structure alignment.

4.6.2 SVM model

Generally the SVM is a margin classifier. It draws an optimal separating hyperplane (decided by w, b) in a high-dimensional feature space between positive examples and negative examples. To avoid over-fitting, the SVM finds the maximum margin hyperplane, the hyperplane that maximizes the minimum distance from the hyperplane to the closest training point. For cases in which no linear separation is possible, the SVM can work in combination with kernel function (indicated by $\phi(x)$) that automatically gives a non-linear mapping to a feature space. The decision boundary is defined by the function:

$$f(x) = \text{sgn}(w \cdot \phi(x) + b) \tag{4.5}$$

Given a new data point x to classify, depending on the sign of the function, the protein alignment could be classified into true and false.

Therefore, given a sequence-structure alignment of two domains in SCOP, if the two domains are from the same family or superfamily, it is counted as positive samples (true), otherwise as negative samples (false). For the

negative samples, the protein pairs at the same fold level are not included. Feature vectors are extracted from the outputs of sequence-structure alignment, which are alignment length, mixed profile length, query sequence length and alignment score. In total, 14,533 pairs are randomly chosen from SCOP to train the SVM.

4.6.3 SVM training and parameters optimization

The SVMlight (Joachims, 1999) is downloaded in this research, which is an implementation of SVM for the problem of pattern recognition. The original code is available at http://www.cs.cornell.edu/People/tj/svm_light/. The SVMlight still has a few adjustable parameters to be determined. The SVM training includes the selection of the proper kernel function parameters and the regularization parameter *C*. Both linear and RBF kernel functions are investigated in this research. The polynomial kernel function is not selected due to its slow training. The result of predicted accuracy with different types of kernel functions is summarized in Table 4.2. The predicted accuracy on test data reached 87.2% with the linear kernel function. However, the accuracy is improved to 90.7% using the RBF kernel function. Thus, the RBF kernel function is used with $\gamma = 5.0$ and *C*=1000 for alignment evaluation.

Kernel function		Predicted accuracy
Linear		87.2%
RBF	$\gamma = 1.0$, C=1000	89.1%
	$\gamma = 5.0$, C=1000	90.7%

Table 4.2 The performance of SVM with different kernel function

The gap penalty parameter (see Section 4.3.3) could also be optimized by increase the predicted accuracy. The results are listed in Table 4.3.

GAP	g							
g'	1.2	1.4	1.6	1.8	2.0	2.2	2.4	2.6
0.0	89.76%	90.01%	90.60%	90.58%	90.44%	90.52%	89.86%	89.95%
0.1	89.98%	90.52%	90.68%	**90.72%**	90.59%	89.89%	90.30%	89.72%
0.2	90.03%	89.87%	90.23%	90.34%	90.23%	90.14%	90.54%	90.21%
0.3	89.95%	89.76%	89.94%	90.45%	89.79%	89.86%	90.28%	89.86%

Table 4.3 The performance of SVM with gap penalty parameter optimization

4.6.4 Neural network model

For comparison purpose, a three-layered BPNN with 4 input neurons (alignment length, profile length, query sequence length, alignment score) and two output neuron (related and unrelated proteins) is also trained for evaluating protein alignment significance, namely MESSM_NN (as shown in Figure 4.10). Six-fold cross-validation test is used for training. The same 14533 pairs are used for NN training as for SVM training. The performance of MESSM_NN is compared with MESSM_SVM (the one with the SVM as confidence evaluation) on benchmark problems. The results are discussed in Chapter 5.

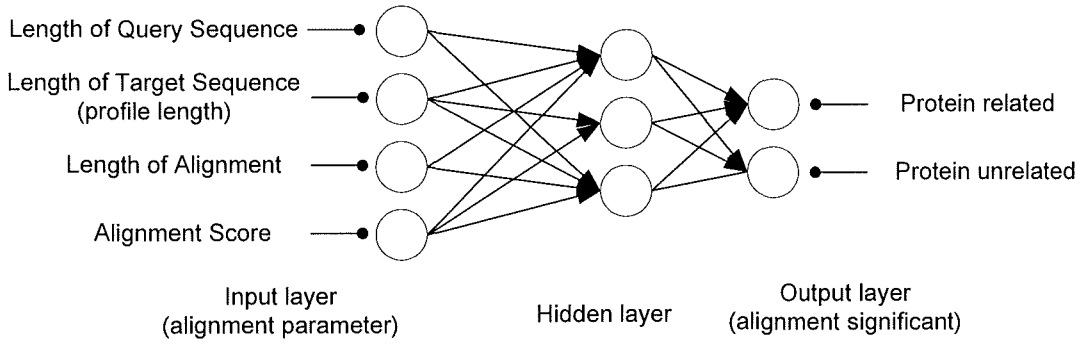


Figure 4.10 The NN model for confidence evaluation

4.7 Summary and discussion

This chapter has presented a new process of developing an efficient tool for protein threading. With a residue level environmental description, the contact measurement is re-adjusted from previous work. A NN is employed to generate structurally-derived substitution mapping rather than the commonly used environment-specific amino acid substitution tables. A mixed substitution scores is proposed by the inclusion of the structurally-derived substitution mapping and the well-developed amino acid substitution matrix. A SVM is used to evaluate the alignment significance. With these three key steps, the new framework for protein threading is developed as an automated method for annotation of genomic sequences.

The performance of the MESSM will be evaluated on four benchmarks, as described in the next chapter. The results will be compared with current threading models based on energy potentials.

CHAPTER 5 EVALUATION OF THE MESSM

To evaluate a protein threading model, several factors should be considered: the method should be fast; it should detect the correct fold near the top with a score of at least moderate significance and it should give a reasonably good alignment. Thus, to verify the performance of the proposed MESSM framework, experiments are carried out on four benchmark data sets. They are: the Fischer et al. (1996) test sets, the ProSup benchmark (Domingues et al., 2000), the Lindahl (Lindahl and Elofsson, 2000) data sets and the Wallner et al. (2004) data sets. The early benchmark of Fischer is used to optimize the μ parameter in mixed substitution score. The alignment accuracy of MESSM is tested by the ProSup benchmark. The Lindahl (Lindahl and Elofsson, 2000) and the Wallner et al. (2004) data sets are used to assess the fold recognition sensitivity. Both the Lindahl and Wallner data sets are designed by Elofsson's group. Several well-established threading methods have been tested on Lindahl's benchmark. The Wallner set has much larger newly-designed data sets with 4,972 proteins. Currently there are no published results on Wallner's data set.

5.1 Fischer's benchmark

5.1.1 Data sets

Fischer's benchmark (Fischer et al., 1996) comprises a variety of structural types. They are 13 α proteins, 25 β , 20 α / β , 7 $\alpha + \beta$ 1 multi-domain and 2 small proteins, as shown in Table 5.1. Each sequence-fold pair is listed according to its type of fold. The lengths of the proteins vary from 62 to 581 residues. Fischer's 68 data sets have very low sequence similarity (below 30%), but with highly similar folds, which is extremely suitable for testing the MESSM.

Fold	Probe	Target	Seq ID
α: 13 pairs			
EF-hand	1osa	4cpv	24
EF-hand	2sas	2scpa	17
Globin-like	1dxtb	1hbg	19
Globin-like	1cpcl	1cola	17
Cytochrome	1c2ra	1ycc	23
Cytochrome	2mtac	1ycc	15
Helix bundle	1aep	256ba	14
4-Helix bundle	1rcb	1gmfa	21
4-Helix bundle	1bbha	2ccya	21
4-Helix bundle	1bgeb	1gmfa	12
DNA-binding (HTH)	1hom	1lfb	19
Peroxidase	1lgaa	2cyp	16
Peroxidase	2hpda	2cpp	18
α/β: 20 pairs			
TIM barrel	1chra	2mnr	20
TIM barrel	2mnr	4enl	18

Fold	Probe	Target	Seq ID
β: 25 pairs			
IG-fold	1pfc	3hlab	22
IG-fold	1tlk	2rhe	24
IG-fold	3cd4	2rhe	25
IG-fold	1cid	2rhe	13
IG-fold	1ten	3hhrb	18
IG-fold	2fbjl	8fabb	22
IG-fold	1fc1a	2fb4h	19
IG-fold	3hlab	2rhe	15
Cupredoxin	1afna	1aoza	19
Cupredoxin	2azaa	1paz	11
Cupredoxin	1aaj	1paz	31
Jelly roll	1caub	1caua	18
Jelly roll	4sbva	2tbva	19
Jelly roll	1bbt1	2plv1	20
Jelly roll	1saca	1ayh	14
Beta propellor	1sim	1nsba	12
Lipocalin	1mdc	1ifc	21
Lipocalin	1mup	1rbp	14

TIM barrel	3rubl	6xia	18	Trypsin	1arb	4ptp	20
Hydrolase	1taha	1tca	16	Trypsin	2snv	4ptp	15
Hydrolase	1crl	1ede	17	Trypsin	2sga	4ptp	21
Thioredoxin	1aba	1ego	21	Trefoil fold	1tie	4fgf	14
Thioredoxin	1dsba	2trxa	13	Trefoil fold	8i1b	4fgf	18
Thioredoxin	1gp1a	2trxa	17	OB fold	1ltsd	1bova	19
Ribonuclease-H	1hrha	1rnh	24	Porin	2omf	2por	17
Actin	1atna	1atr	15	α+β: 7 pairs			
Open sheet	1npx	3grs	20	UB fold	1fxia	1ubq	18
Open sheet	2cmd	6ldh	23	Alpha + beta	2sara	9rnt	12
Open sheet	1gky	3adk	24	Ribonuclease	1onc	7rsa	26
Open sheet	1eaf	4cla	21	SH2	2pna	1shaa	29
Open sheet	1gal	3cox	18	Ferredoxin	5fd1	2fxb	21
Open sheet	2pia	1fnr	18	Monellin	1cew	1mola	10
Open sheet	2gbp	2liv	16	Monellin	1stfi	1mola	8
Open sheet	3chy	4fxn	14	Multi-domain and small proteins: 3 pairs			
Open sheet	1mioc	1minb	16	Small	1isua	2hipa	16
Open sheet	1ak3a	1gky	24	Small	1hip	2hipa	19
				Mixed	2hhma	1fbpa	13

Table 5.1 Fischer's 68 benchmark pairs. Fold, query sequence's type of fold; Probe, query sequence; Target, expected match protein; Seq ID, percentage identical residue in sequence between probe and target.

5.1.2 Results

5.1.2.1 Optimisation parameter μ in combined substitution score

For the MESSM approach proposed in this research, there is an adjustable parameter (μ) in the substitution score (function 4.3, page 94). Since the parameter μ lies in the range of 0~1, the simple grid search method is adopted. The procedure of optimization is as following:

1) Divide the zone [0, 1] into a coarse grid of trial parameters; compute a combined substitution score with each trial parameter.

2) Test Fischer's data sets on the MESSM model: For each target sequence in Fischer's data, a correct hit is achieved if the MESSM model ranks the correct matching template protein at the first rank. The MESSM model is tested on this benchmark with each substitution score by dynamic programming. The number of correct hits is counted for each trial parameter.

3) Look for the region that seems to contain the maximum number of hits and zoom in on it.

4) Repeat step 1 through step 3 but with a smaller range and a finer grid.

5) Stop the optimisation when there is no more improvement on the maximum number of correct hits by Fischer's benchmark.

Three levels of success are defined based on the number of Fischer's pairs in the top 1, top 5 and top 10 positions. As shown in Figure 5.1, the number of correct hit with sequence substitution only (μ =0) and structural derived substitution mapping only (μ =1) are also computed.

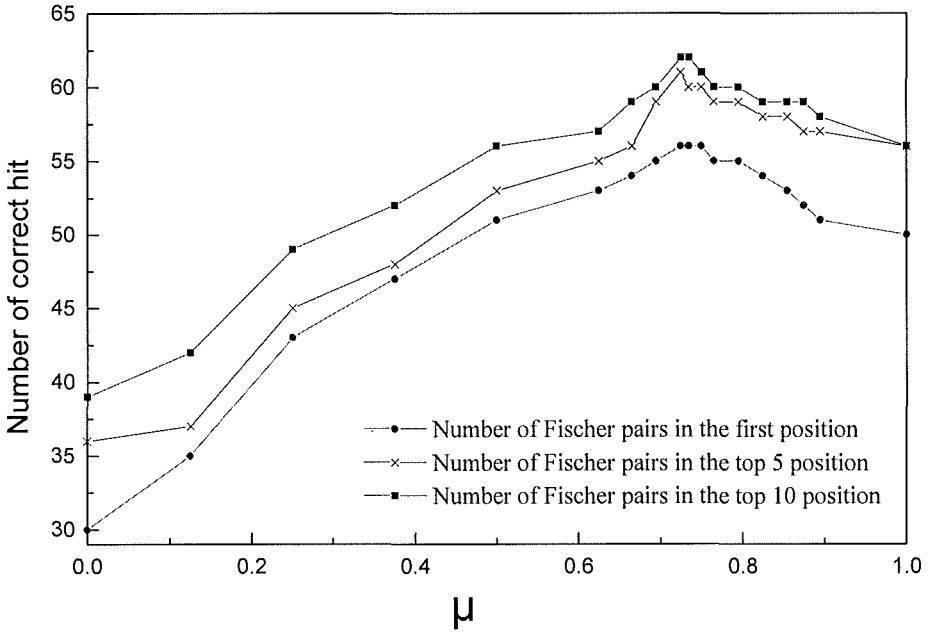


Figure 5.1 the number of hit by Fischer's benchmark with different μ

In Figure 5.1, the fold recognition model with sequence substitution only (BLOSUM30, μ=0, called SSM) could correctly identify 30/36/39 Fischer pairs in the top 1/5/10. Whereas the model with the structural substitution derived mapping (μ =1, called ESM) could successfully identify 50/56/56 Fischer pairs in the top 1/5/10. It demonstrated that, due to the structural information extracted by NNs, the performance of the ESM is far better than the SSM. Figure 5.1 also showed that a peak of performance was observed around the values of parameter $0.70 < \mu < 0.75$. The improvement over both the ESM and the SSM is significant up to about 38%. The highest success rate for the proposed MESSM on Fischer's data sets was 56/68 when μ=0.725.

5.1.2.2 Recognition performance

Several well-established fold recognition methods were also tested on Fischer's benchmark. For example, PSI-BLAST (Altschul et al., 1997) is a well-known sequence alignment method; GenTHREADER (Jones, 1999) use a classical sequence alignment algorithm to generate query-template

alignments, and then evaluates the alignments by a threading potential. It provides a confidence measure for each predicted fold recognition using a NN; COBLATH (Shan et al., 2001) uses a combined approach of PSI-BLAST and threading techniques to fold recognition; SPARKS (Zhou and Zhou, 2004) is a fold recognition model built on a knowledge-based energy score combining with sequence-profile and secondary structure information. The testing results of MESSM are compared with them and listed in Table 5.2. It shows that the proposed MESSM has the same performance as that of COBLATH and SPARKS, but it is worse than a computationally more intensive, hierarchical threading method called PROSPECTOR (Skolnick and Kihara, 2001), which could correctly recognize 58-61 out of 68 pairs.

Method	Number of correct hits
PSI-BLAST[a]	41
GenTHREADER[b]	50
COBLATH[c]	56
SPARKS[c]	56
PROSPECTOR[a]	58-61
MESSM_SVM[d]	56
MESSM_NN[e]	53

Table 5.2. Performance of different methods for fold recognition on Fischer's benchmark

a Result from Skolnick and Kihara (2001).

b Result from Jones (1999).

c Result from Zhou and Zhou (2004).

d This is the proposed framework with SVM for confidence evaluation.

e This is the proposed framework with NN for confidence evaluation.

MESSM_SVM is the proposed framework with a SVM trained for confidence evaluation. For comparison purpose, a NN with a six-fold cross validation approach is also trained to predict the alignment

significance. The same 14533 pairs as training data for the SVM are chosen from SCOP to train this NN. The comparison results between MESSM_SVM and MESSM_NN are shown in Table 5.2. It is clear that the MESSM_SVM performs better than MESSM_NN. Thus, SVMs are shown to be superior to NNs for this specific problem. In the following section, MESSM refers to the proposed model with the SVM for confidence evaluation (MESSM_SVM).

5.2 Alignment Accuracy Test with ProSup benchmark

In a fold recognition program, a target sequence is aligned with all structures in a fold library. Incorrect alignments may result in unfavorable scores and a failure to recognize relationships among proteins. Also, structural models derived from incorrect alignments might be misleading in subsequent structural and function studies. Therefore, correct alignments are fundamental for the success of the fold recognition techniques. Generally it is found that fold recognition method produce very inaccurate alignment when protein pairs have very low sequence similarity (Jones, 1999).

ProSup benchmark (Domingues et al., 2000) was prepared by Sippl's group to test the alignment accuracy of fold recognition methods. It can be found publicly available at: http://lore.came.sbg.ac.at/Services/Benchmark/Prosup/. The ProSup data set consists of 127 pairs of proteins derived from PDB. These pairs of protein have clear structure similarity and no pairs have a sequence identity greater than 30%. The correct alignments for these pairs are obtained by the structural comparison program ProSup. The accuracy of an alignment was obtained by calculating the percentage of matches between the correct alignment and the alignment made by a fold

recognition method. Since generally the structure-derived alignments (correct alignments) have multiple solutions, the percent of matches is the maximum value obtained from a comparison to all alternatives.

Suppose N is the number of protein pairs in the ProSup benchmark, L_i is the number of residue pairs in the correct alignment, A_i^{exact} is the number of residue aligned exactly the same as in the structural alignment (ProSup), the alignment accuracy of each pairs is $\sigma_i = A_i^{exact} / L_i$. The average percentage of correctly aligned residue per protein pair is $\sigma = \sum_i \sigma_i / N$.

Table 5.3 compares the alignment performance of MESSM with several other methods. The structural alignment results of ProSup benchmark downloaded in this research is updated by 18/Jan/2001. Several published methods before year 2001 listed in Table 5.3 may adopt different sequence and structure database. Although this is not a strict comparison, it can serve as an approximate indicator for the accuracy of the MESSM. In Table 5.3, a significantly better performance than the current models indicates that the MESSM method is promising to provide a more accurate fold-recognition alignment. This result is consistent with a previously reported study (Jones, 1999) that the sequence-profile alignment algorithms that utilize the profile information can generate reasonably good alignments among the remotely related proteins.

Method	Accuracy (%)
PSI-BLAST[a]	35.6
FASTA[b]	31.4
Sequence[b]	34.1
Threading[b]	48.0
SPARKS[a]	57.2
PROSPECT II [c]	57.7
MESSM[d]	**59.7**

Table 5.3 The average alignment accuracy for ProSup benchmark per pair of proteins

a Result from Zhou and Zhou (2004).

b Result from from Domingues et al. (2000).

c Result from Kim et al. (2003).

d This is the proposed framework.

5.3 Lindahl benchmark

5.3.1 Data sets

The Lindahl set (Lindahl and Elofsson, 2000) was designed to assess the recognition performance of protein fold recognition algorithms. It was created from a subset of the SCOP version 1.37. It has 976 proteins where no two proteins have more than 40% sequence identity. There are 555, 434 and 321 pairs of proteins in the same family, superfamily and fold, respectively. (Proteins sharing a family have a "clear evolutionary relationship"; those within a superfamily are of "probable common evolutionary origin"; while the fold level is characterized by "major structure similarity".) The complete benchmark is available from:

http://www.sbc.su.se/%7Earne/protein-id/. The fold recognition method is tested by checking whether or not the method can recognize the member of same family, superfamily or fold as the first rank or within the top five ranks.

5.3.2 Results

The performance on all against all comparisons of Lindahl's 976 sequences is measured at three different similarity levels: family, superfamily and fold. The results of MESSM are summarized in Table 5.4 and compared with several well-established methods. FUGUE (Shi, et al., 2001) and SPARKS (Zhou and Zhou, 2004) represent two of the better performing threading programs currently available. Table 5.4 shows that the performance of MESSM is better than THREADER (Jones, 1999) and PSI-BLAST (Altschul et al., 1997). The overall performance of the MESSM is similar to FUGUE and SPARKS. MESSM performs better on the number of fold as first rank, worse on others.

FUGUE and SPARKS are two elaborate designed methods with multiple sequence alignment information integrated with threading techniques. Multiple sequence alignments can provide the identification of conserved sequence regions, which reveal the evolutionary information of proteins. Proteins in family level have a clear evolutionary relationship and proteins in superfamily level may have a common evolutionary origin. With the multiple sequence alignment information included in the FUGUE and SPARKS, the two models perform better than MESSM on family and superfamily level. Unfortunately, the current framework of MESSM didn't include multiple sequence alignments. It is hope that by adding the multiple sequence alignment information into the MESSM (see Section 7.2 future work), MESSM can outperform FUGUE and SPARKS on family and superfamily level as well.

Method	Family Only		Superfamily Only		Fold Only	
	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5
THREADER[a]	49.2%	58.9%	10.8%	24.7%	14.6%	37.7%
PSI-BLAST[a]	71.2%	72.3%	27.4%	27.9%	4.0%	4.7%
FUGUE[a]	82.2%	85.8%	41.9%	53.2%	12.5%	26.8%
SPARKS[a]	81.6%	88.1%	52.5%	69.1%	24.3%	47.7%
MESSM[b]	**76.87%**	**83.36%**	**51.52%**	**65.34%**	**25.23%**	**45.48%**

Table 5.4. Performance of different method for fold recognition on Lindahl benchmark.

a Result from Zhou and Zhou (2004).

b Our proposed framework.

Table 5.4 gives the percentage of correct matches, but it does not tell the reliability of the match. For example, a match could be the top rank but have a very low score as long as all others have even lower scores. Therefore, sensitivity-specificity plots (Rice and Eisenberg, 1997; Lindahl and Elofsson, 2000) are drawn to measure the reliability of the match. Given a threshold value, the sensitivity is defined as:

$$SENS(threadhold) = \frac{TP(threshold)}{TP(threshold) + FN(threshold)} \tag{5.1}$$

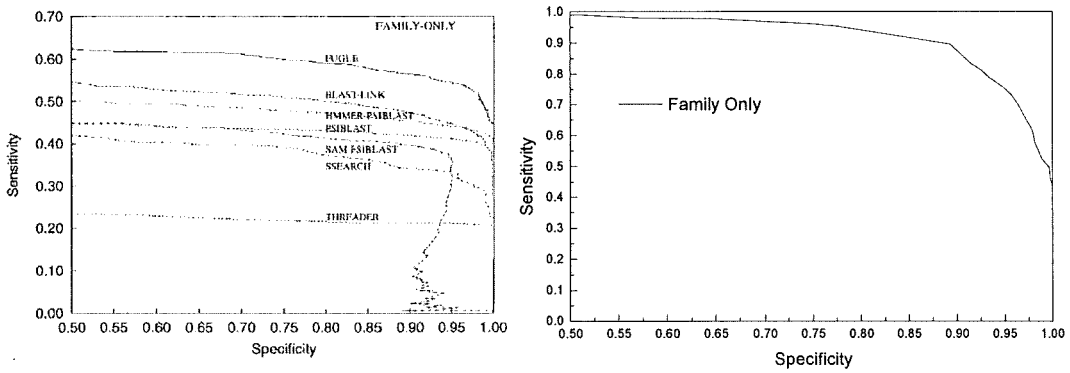
where $TP(threshold)$ is the number of correct hits having a score above the threshold; $FN(threshold)$ is the number of correct hits with a score less than the threshold. The specificity is defined as:

$$SPEC(threadhold) = \frac{TP(threshold)}{TP(threshold) + FP(threshold)} \tag{5.2}$$

where $FP(threshold)$ is the number of false hits that have a score above the threshold. The specificity measures the probability that a pair of sequences

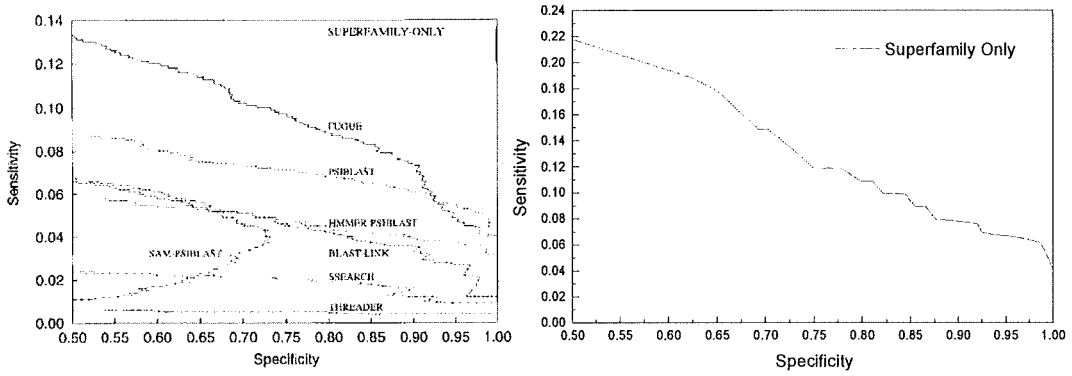
with a score greater than a certain threshold really is a true hit. The sensitivity is plotted as a function of specificity, each point corresponding to a certain threshold.

The sensitivity-specificity curves of the Lindahl benchmark are drawn in Figure 5.2, Figure 5.3 and Figure 5.4. For comparison purpose, the results of other fold recognition models are also presented in the three Figures. At the family level, the MESSM obtained a sensitivity of 52% at 99% specificity, while the best performance of the other methods, was obtained by FUGUE, hit 49% sensitivity at 99% specificity (Figure 5.2). In Figure 5.3, MESSM recognized 5.6% of homologous pairs at the superfamily level with high confidence (99% specificity). At 50% specificity, MESSM achieved 21.7% sensitivity at superfamily level. In contrast, none of the methods compared was able to achieve sensitivity of more than 5% at 99% specificity. FUGUE achieved 4% and 13% sensitivity respectively at the same specificity level. The results at fold level reveal that none of the current methods are capable of reliably recognizing the similarity between two proteins that have major structural similarities only (Figure 5.4). However, the MESSM shows better performance in both ranking protein pairs at the top and the specificity-sensitivity curves than other fold recognition models. MESSM could achieve 17% sensitivity at 50% specificity.



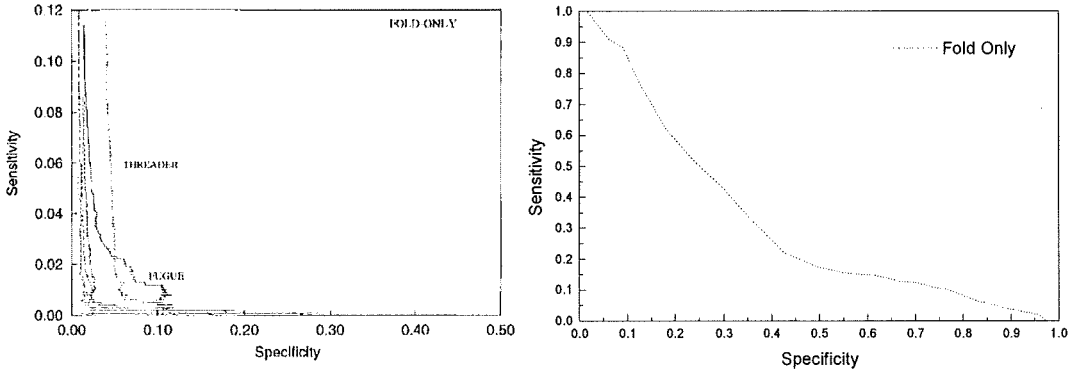
(a) (b)

Figure 5.2 specificity-sensitivity curves using Lindahl's benchmark on family level: (a) The results of other fold recognition models from Shi et al.'s (2001) paper; (b) The performance of the proposed MESSM model.



(a) (b)

Figure 5.3. specificity-sensitivity curves using Lindahl's benchmark on superfamily level: (a) The results of other fold recognition models from Shi et al.'s (2001) paper; (b) The performance of the proposed MESSM model.



(a) (b)

Figure 5.4. specificity-sensitivity curves using Lindahl's benchmark on fold level: (a) The results of other fold recognition models from Shi et al.'s (2001) paper; (b) The performance of the proposed MESSM model.

The specificity-sensitivity curves offer an overview of the quality of confidence score in the MESSM model. For example, for all those homologous proteins that come out as the top rank in superfamily level, the MESSM can recognize 5.6% of homologous protein pairs confidently, whereas FUGUE can only recognize 4% with 99% confidence. Though MESSM could not identify more correct matches in the top rank compared with other methods, the quality of the confidence score is highly improved. This improvement is highly likely to be caused by the employment of SVM in the MESSM, which derives an apparently highly reliable score function.

5.4 Wallner's benchmark*

* This benchmark test is suggested and provided by Dr. Arne Elofsson from Stockholm Bioinformatics Center, Stockholm, Sweden.

5.4.1 Data sets

The Wallner's benchmark (Wallner et al., 2004) is a significantly large and newly well-created data set. The data set is built on a subset of SCOP (version 1.57) in which no protein domains have more than 75% sequence identity to any other member of data set. It contains 4972 proteins whose domains from SCOP class *a* to *e* (ignoring membrane protein, small proteins, coiled-coiled proteins, low-resolution structures, peptides and designed proteins). The detailed number of Wallner's benchmark is shown in Table 5.5.

Description	Number
Number of protein domains	4,972
Number of different families	1,543
Number of different superfamilies	905
Number of different folds	579
Number of pairs on family level	52,532
Number of pairs on superfamily level	101,954
Number of pairs on fold level	125,090

Table 5.5 Description of the Wallner's benchmark set (Wallner et al., 2004)

Though the MESSM is a threading model, it should also detect all protein domain pairs with >30% sequence identity. The Wallner's benchmark data set is not affected to test the performance on superfamily and fold level, because there are no proteins from two different families with >30% sequence identity. Thus, this benchmark set is suitable for verifying the MESSM.

5.4.2 Evaluation results

The identified pairs at different similarity levels are shown as top ranks listed in Table 5.6. The sensitivity-specificity curves are drawn in Figure

5.5 and 5.6. For comparison purpose, the results of SSM (the fold recognition model with sequence substitution only) and ESM (the model with the structural substitution derived mapping only) are also computed. Figure 5.5 and 5.6 demonstrate that with the structural information extracted by NNs, the model ESM could obtain a better performance than the SSM with sequence substitution only. MESSM also performs better than both ESM and SSM. At the family level, the MESSM model obtains a sensitivity of 72% at 99% specificity, whereas ESM achieves a sensitivity of 69% and SSM achieves a sensitivity of 63% at 99% specificity respectively. At the superfamily level, the MESSM model obtains a sensitivity of 34% at 90% specificity. In contrast, ESM achieves a sensitivity of 27% and SSM achieves a sensitivity of 19% at 90% specificity respectively. Figure 5.6 shows that at fold level, MESSM achieves a sensitivity of 20%, whereas ESM achieves a sensitivity of 18% and SSM achieves a sensitivity of 14% at 90% specificity respectively. Our results are compared with the best results of profile-profile method reported by Wallner et al. (2004), which have a sensitivity of 72% at 99% specificity on family level and a sensitivity of 22% at 90% specificity on the superfamily level. Though this is not a strict comparison due to the different confidence-evaluation method used by each model, it shows that the MESSM model has a good performance on protein fold recognition.

Method	Family Only		Superfamily Only		Fold Only	
	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5
MESSM	74.94%	76.37%	54.03%	64.86%	20.03%	35.92%

Table 5.6. Performance of MESSM on Wallner's benchmark (identified pairs at different similarity level)

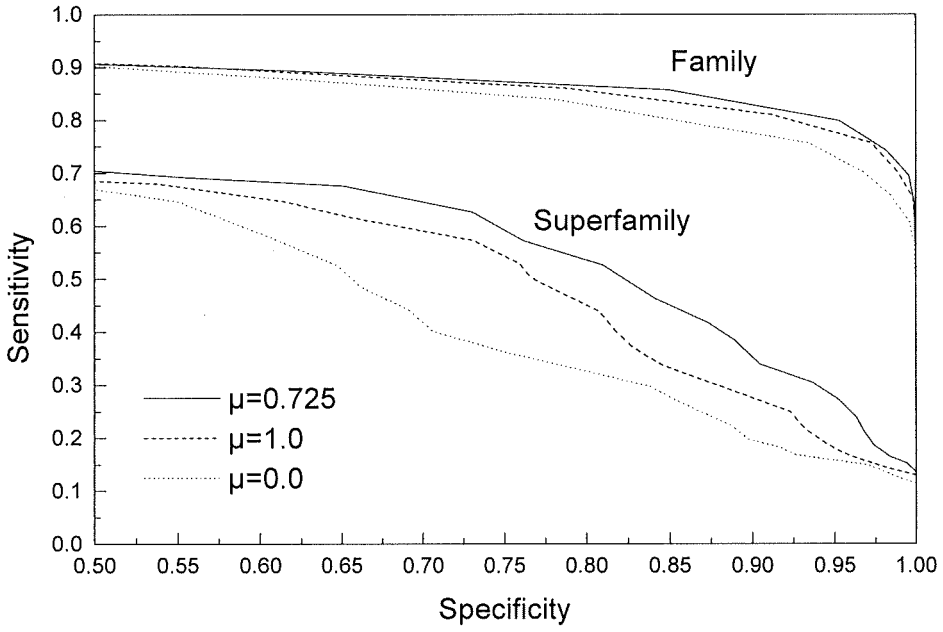


Figure 5.5. Family and superfamily level specificity versus sensitivity curves on the Wallner's benchmark.

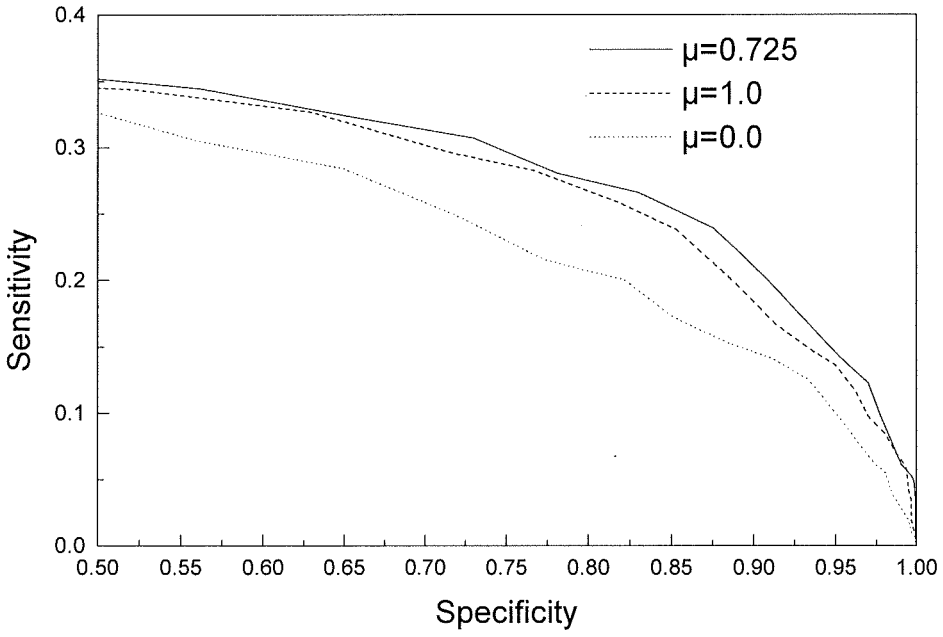


Figure 5.6. Fold level specificity versus sensitivity curves on the Wallner's benchmark.

5.5 Discussion

MESSM is a threading framework with sequence-profile alignment and the SVM as a significant assessor. It is a fast program. It could make an alignment between probe sequence (150 amino acids) and a profile of 4775 template proteins in 30 seconds on a PC with 1G memory Pentium IV. In MESSM, the protein representative fold library is predefined as $n \times 20$ matrices. Once the whole library is loaded into the PC memory, the computational time is mainly the searching time for global-local alignment algorithm. Thus, MESSM is less computationally expensive and fast program.

Tested on four benchmark problems, MESSM shows comparable performance on protein fold recognition to those more computational intensive, energy potential based fold recognition models. The quality of the score function is improved compared to the current models. The alignment accuracy is better than those models. The improvements are due to the three key features imported in the MESSM framework. Currently, the MESSM is a simple model. It presents a new process to develop an efficient tool for protein fold recognition. By considering secondary structure and integrating multiple alignments into the current model of MESSM, a further improvement on the MESSM model is expected.

In the following Chapter, this research is extended to build a threading score by following the contact potential approach.

CHAPTER 6 PROTEIN DECOY AND NATIVE DISCRIMINATION BY THREADING SCORES (CONTACT POTENTIAL APPROACH)

Threading using the contact potential approach differs from the structural profile approach. It considers a detailed network of pairwise interactions between individual residue rather than just assigning them to a basic environmental class (Jones and Hadley, 2000). In general, the most successful protein threading methods are based on contact potential techniques (Xu and Xu, 2000; Kim et al., 2003), although contact potential approaches generally require more computational cost than the structural profile approach.

In this project, besides the work of MESSM based on structural profile technique (Chapters 4 and 5), a study of protein threading is extended to the contact potential approach by using the new residue contact measuring scheme developed in MESSM. As shown in the right part of the Figure 3.2, the design and evaluation of a threading score (TES; Threading with Environment-specific Score) will be carried out and reported in this Chapter. The threading score is tested by discrimination of the protein native structure from decoys.

6.1 Introduction

Protein threading through the contact potential approach always contains a scoring function formulated in terms of knowledge-based potentials. For such a knowledge-based potential, generally, the statistical analysis of known protein structure is used to measure the free energy between the interaction of residues or atoms. As a result of such analysis, the so called contact energies are used to evaluate the protein sequence-structure fitness.

In this Chapter, a model named TES (Threading with Environment-specific Score) is developed to build a new threading score function with the use of ANNs. The TES model is constructed on the basis that each amino acid residue in a protein tertiary structure stays in a particular structural environment. Since different protein sequences may adopt the same fold, different amino acid residues may stay in a similar structural environment. The focus of this method is on the environment surrounding an amino acid residue in a protein structure and how this environment serves to determine the identity of that residue without the measurement of the free energy between the interactions of residues commonly used in pairwise contact potentials. Thus, given a protein structure with a residue level environment description, the compatibility of residue in sequence with its structural environment is presented. A threading score is constructed by log-odds scores of predicted probabilities from the trained NN to determine which residue best fits its environment.

Differing from the TUNE model proposed by Lin et al. (2002) which encoded the contacts between residue side-chain and its neighbours as the overlapping volume of their contact regions, the TES model is built on the new contact measuring scheme developed in the MESSM. Two

computational experiments are carried out to verify the TES model on discrimination of protein decoy and native structures. The results are compared with knowledge-based energy potentials and TUNE model in Section 6.3.

6.2 Data and Methods

6.2.1 Description of structural environments

The representation of the protein is an essential component in a protein threading program. The representation of the protein structure can be an all atom structure, a backbone structure, a string of β carbon atoms, a set of inter-residue distances or a string of amino acid names. It was demonstrated that the efficiency of a potential energy function depends on the degree of the details of the structural description. The atom distance-dependent pairwise potential has been shown to be more accurate than those with residue-based potential (Samudrala and Moult, 1998; Lu and Skolnick, 2001) but with a higher computational cost.

In this research, the same residue level environmental description of proteins and contact measuring scheme are adopted as the one used in MESSM model (Chapter four). Each amino acid residue is described using main chain and pseudo side-chain spheres; residue neighborhood and contact are built on the fact that if the space between two amino acids is larger than one water molecule or a third residue, then they are considered to be too far to have contact. Thus, two kinds of contacts are considered. They are side-chain to side chain contact and side chain to main chain contact.

6.2.2 *Neural network model*

Reported by Lin et al. (2002), NNs are very well suited for mapping the probabilities of observing each amino acid residue in its structural environment. In this research, a standard one hidden layer feed-forward BPNN is adopted for protein sequence-structure mapping. Environment-specific sequence-structure compatibility is captured by the NN model. A log-odds score of predicted probabilities from the trained NN model is constructed to determine which residue in the sequence best fits its environment.

A three-layered fully connected back-propagation feed-forward NN with 25 input neurons, 20 output neurons and 22 hidden neurons is used to predict the probabilities of observing different amino acid type in a structural environment. As shown in Figure 6.1, a total of 25 input neurons represent the features of the structural environment of each amino acid residue on the protein sequence chain. One input unit is used for residue solvent ability, measured by the sum of all the contacts. Four units are used to represent the distances from the alpha carbon to the alpha carbons, describing the local structure. Based on twenty types of amino acids, the twenty inputs that remain are employed to encode the neighbour contacts. For each neighbour, a value of one is added to the corresponding unit according to its amino acid type. A weight w_{ij} (see formula 4.1) is added to reflect the influence of the neighbour contact. In this work, various network architectures are tested by changing the number of neurons in the hidden layer from 10 to 30. The 22-hidden-neuron model is selected due to its lower training error. The targeted output of the NN is the amino acid type in the structural environment, which is encoded by the orthogonal encoding scheme (20 units).

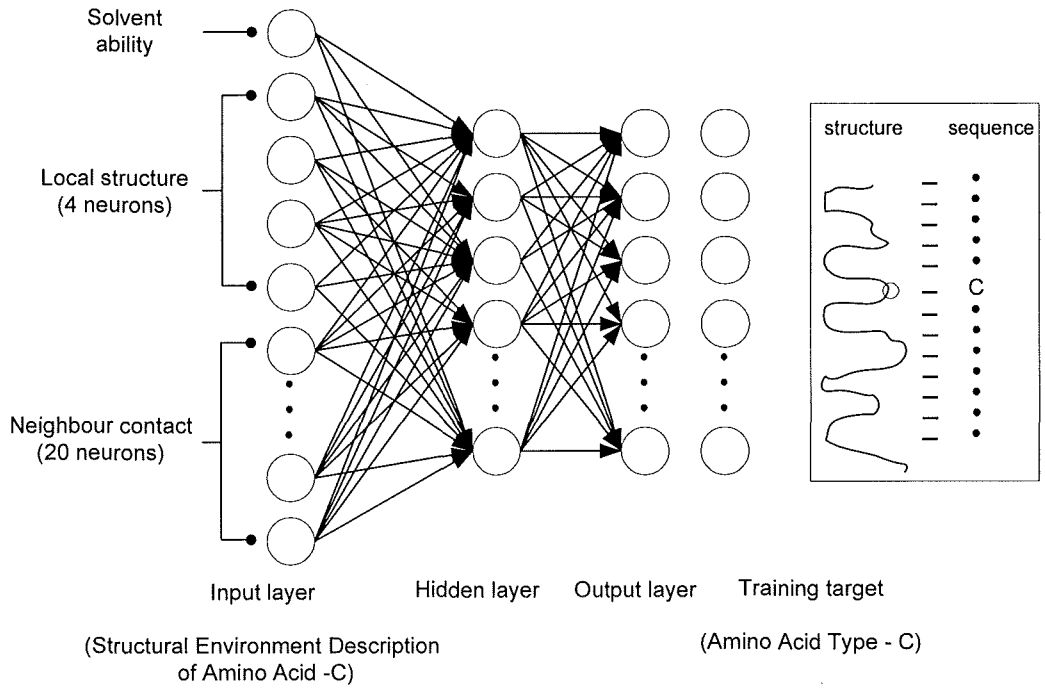


Figure 6.1 The NN model for training (structure-sequence mapping). Three layered feedforward NN: 25 input neurons describing the structural environment of an amino acid, 22 hidden neurons and 20 output neurons.

The standard logistic sigmoid activation function is used for the hidden layer and the softmax activation function for the output layer due to the output range (0 to 1). The relative entropy error is used to measure the performance of NNs (Baldi and Brunak, 2001). Each input value was scaled to be in the range of 0~1 using the function:

$$Input = \frac{1}{1 + e^{-a(x-b)}} \tag{6.1}$$

where x is the raw input value and a and b are constants. a is given to make the average input value to be zero and b is chosen to scale most of the input into the range of 0~1. In this research, a=1 and b=10 (McGuffin and Jones, 2003).

A back-propagation algorithm is employed to minimize the mean difference between the predictions and the real amino acid types. A ten-fold cross-validation approach was used in the training. The NN is trained with 10~20 different starting conditions, that is, the same ten-fold cross-validation experiments were run 10~20 times with random initial weights and biases. After training, the performance of the NN is tested using an unseen test set. The NN with the least error is chosen as the appropriated model for the evaluation. The training results are shown in Appendix III.

6.2.3 The back-propagation neural networks model to calculate environment-specific score

Let P (x | E) be the frequency of observing residue x in an environment E. Given a residue in a protein structure, its structure environment is encoded and entered as the input to the trained three-layered NN. The output is the prediction of the probability P(x | E). A log-odds score of the compatibility is given by (Rice and Eisenberg, 1997; Lin, et al., 2002):

$$S = \ln(\frac{P(x \mid E)}{P(x)}) \tag{6.2}$$

where $P(x)$ is the occurrence probability of the residue x in the sequence. The higher the logarithm likelihood score is, the better x residue fits its structure environment E.

6.2.4 Datasets

The structure classification database of SCOP (v1.611, September 2002) is used to select training and testing data sets for the NNs. From the 2900 protein family representatives, whose sequence identity are less than 40%, the structural environment description of each amino acid is obtained for

535,525 residues. Nine tenths of all the domains (2610 domains, 487,328 residues) are randomly selected for training, the remaining one tenth domains (290 domains, 48,197 residues) are used as testing data for NN.

6.3 Experiments and Results

For residue-based potentials, two kinds of measurement are typically used: z-scores for gapless threading, and the ability to discriminate native structures from decoys (Lu and Skolnick, 2001). A decoy set includes the near-native conformations of a protein together with a large ensemble of misfolded models. In this section, two experiments with the benchmark are carried out to verify the performance of our TES model by discriminating native structure from decoys. First, three early decoy sets are selected from the PROSTAR website (http://prostar.carb.nist.gov/) to verify the accuracy of the TES model. They are *asilomar* (CASP2) (Mosimann et al., 1995), *misfold* (EMBL_misfold) (Holm and Sander, 1992) and *ifu* (Unger and Moult, 1991). Then seven decoy sets obtained from the Decoys 'R'Us (http://dd.stanford.edu/) are adopted to evaluate the performance of TES. The two evaluations are worked on PCIV Linux system using C++ and Perl language.

For each structure of evaluation data, the sum of the compatibility scores of every residue is calculated. If the summed compatibility score of the native structure is higher than that of the decoys, it demonstrates that the TES model performed correctly in the discrimination of these native-decoy pairs of protein.

6.3.1 PROSTAR decoy sets

For comparison with other published works, three early decoy sets of *misfold, asilomar* and *ifu* are chosen from PROSTAR website. The *misfold* decoy set is generated by Holm and Sander (1992), consists 24 pairs of proteins with the same number of residues in the chain, but different sequences and conformations. The *asilomar* decoy set is formed by 41 comparative models of six different proteins (Mosimann et al., 1995) in the first experiment on the Critical Assessment of protein Structure Prediction methods (CASP2). The *ifu* decoy set is based on a set of 44 peptides which are proposed to be independent folding units as determined by local hydrophobic burial and experimental evidence (Unger and Moult, 1991).

Using the trained NN, the summed environment-specific compatibility score for the each structure in the decoy sets of *Asilomar, misfold* and *ifu* are computed. The results are shown in Appendix IV. If the compatibility value of decoy is less than that of native protein, then it means the model could successfully distinguish the native and decoy protein and vice versa. In the *misfold* subset, the proposed model selected the native structure 100% correctly from decoy structure. In the test set of *asilomar*, the proposed model failed to pick out 7 of 41 test sets. For the *ifu* decoy set, the model missed 10 of 44 native structures. So, of the total 109 structure-decoy pairs, the proposed model successfully detected 92 pairs (Table 6.1). In Table 6.1, the performance of the TES is compared with potential based models (DFIRE-A, KBP, RAPDF, RKBP and CDF) and TUNE model. Though the overall performance of the TES is not better than KBP, which is the mean force potentials with atomic structure description, it is comparable to it. Table 6.1 shows that the performance of TES is better than those residue contact potentials, like RKBP and CDF. Also TES is better than the TUNE model which is also a NN model but with different residue contact description.

Set \ Method	DFIRE-A[a]	KBP[b]	RAPDF[c]	RKBP[d]	CDF[c]	TUNE[e]	**TES[f]**
misfold	24/24	24/24	24/24	24/24	19/24	24/24	**24/24**[g]
asilomar	-	37/41	37/41	35/41	35/41	31/41	**34/41**
ifu	34/44	32/44	30/44	22/44	21/44	31/44	**34/44**
Total	-	93/109	91/109	81/109	75/109	86/109	**92/109**

Table 6.1 Evaluation of proposed model TES with other published potentials on decoy sets from PROSTAR

[a] DFIRE-A is the mean-force atomic potential from Zhou and Zhou (2002).

[b] The atomic KBP is the atomic potential developed by Lu and Skolnick (2001).

[c] RAPDF and CDF are atomic and residue-based potentials respectively, from Samudrala and Moult (1998).

[d] RKBP is a residue-based quasichemical potential from Skolnick et al.(2000).

[e] TUNE is ANN model from Lin et al. (2002).

[f] This is the proposed model in this research.

[g] The first number in each cell is the number of correctly identified decoys, and the second number is the total number of decoys. The first column lists the subsets of decoys.

6.3.2 Decoys'R'Us

The seven decoy sets obtained from the Decoys 'R' Us database are as follows:

1) The 4state_reduced set which is generated using a four-state off-lattice model together with a relaxation model. This decoy set consists of seven small protein sequences, each with ~600-700 decoys whose RMSD (root-mean-square-deviation) ranges from 0Å (native structure) to 10Åfrom the native structure (Park and Levitt, 1996). All the decoy structures in this set have the native secondary structures (Lu and Skolnick, 2001).

2) The fisa set which contains decoys from four small alpha-helical proteins. The main chains for these decoys were generated using a fragment insertion simulated annealing procedure to assemble native-like structures from fragments of unrelated protein structures with similar local sequences using Bayesian scoring functions (Simons et al., 1997).

3) The fisa_casp3 set, which contains decoys of proteins predicted by the Baker group for CASP3 using the same method as in fisa set (Simons et al., 1997).

4) The hg_structal set, which contains decoys for 29 globins. Each globin has been built by comparative modelling using 29 other globins as templates with the program segmod (Samudrala et al., 1998).

5) The lattice_ssfit set, which contains conformations for eight small proteins generated by ab initio methods (Samudrala et al., 1999; Xia et al., 2000).

6) The lmds (Local minima decoy set), which contains decoys that were derived from the experimental secondary structures of ten small proteins that belong to diverse structural classes (Samudrala and Levitt, 2000).

7) The semfold set which contains six proteins generated at CASP4 by incorporating multiple functions and uses hierarchical filtering to reduce the number of conformations from a large sample to a tiny fraction to enhance the signal and eliminate false positives (Samudrala and Levitt, 2002).

Previous studies of scoring functions all tried to correlate RMSD with energy scores although the relationship between the two is less than clear. So, in this study, the correlation coefficients between the RMSD (root-mean-square-deviation) and the environment-specific score are computed for seven decoy sets from Decoys'R'Us.

Taking the PDB code 1ctf from 4state_reduced as an example, the summed environment-specific score for the each structure in the decoy set is computed with the trained NN model first. Some example results are shown in Table 6.2. The last one in the Table 6.2, which has zero RMSD value, is the native structure of 1ctf and has the largest compatibility score.

Number[a]	RMSD(Å)[b]	Compatibility[c]
1	6.441	15.2985
2	4.581	10.8786
3	4.52	8.66074
4	4.471	5.07709
5	5.871	9.66547
6	2.793	19.1413
7	4.084	21.8528
8	6.522	-12.8234
9	5.592	-3.69314
10	5.773	12.1679
...	...	...
...	...	...
625	4.534	21.9827
626	1.67	28.7662
627	5.284	10.8742
628	7.06	-2.78423
629	2.096	27.0661
630	6.525	8.06203
631	0	35.9776

Table 6.2 Example results of compatibility vs. RMSD of PDB code --1ctf from 4state_reduced decoy sets

[a] The number of proteins. For the case of 1ctf, there are totally 630 decoy sets with one native protein.

[b] Root-mean-square-deviation. In this case, the last one with zero RMSD is native protein.

[c] The compatibility value of each protein, which is calculated by adding all the compatibility scores of every residue in protein sequence. In this example, if the compatibility value of the last one (native protein) is the largest value in this column, then it means the proposed TES model could successfully distinguish native and decoy proteins.

Then, the correlation coefficients between the RMSD and the environment-specific score are computed with the statistical formula 6.3.

$$r = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{\sigma_x} \bullet \frac{y_i - \bar{y}}{\sigma_y}\right) \tag{6.3}$$

where $\bar{x}$ is the average of x_i and σ_x is the standard deviation of x_i. (y is the same).

Following the example above, the correlation coefficients are computed for all the seven decoy sets and the results are presented from Table 6.3 to Table 6.9. Among all the decoy sets, only the 4state_reduced data set and hg_structal data set show strong correlations between the RMSD and the environment-specific score. In Table 6.3 and Table 6.6, the RMSD-environment specific score correlation coefficients for 4state_reduced and hg_structal sets range from 0.27 to 0.91. For the other decoy sets, as shown in Table 6.4, Table 6.5, Table 6.7 ~ Table 6.9, the correlation coefficients only range from 0.02 to 0.41. The performance of TES model is in significant differences between the decoy sets. It may be due to the different generation methods used for those decoys. These results are in line with studies reported by others (Zhou and Zhou, 2002).

PDB ID	Description	Decoy number	Residue number	RMSD range	R
1ctf	C-terminal domain of ribosomal protein L7/L12	631	68	2.2-10.2	0.623
1r69	N-terminal domain of phage 434 repressor	676	63	2.3-9.5	0.555
1sn3	Scorpion toxin variant 3	661	65	2.5-10.5	0.419
2cro	Phage 434 Cro Protein	675	65	2.1-9.7	0.437
3icb	Vitamin D-dependent calcium-binding protein	654	75	1.8-10.7	0.685
4pti	Trypsin inhibitor	688	58	2.8-10.8	0.443
4rxn	Rubredoxin	678	54	2.6-9.3	0.586

Table 6.3 The correlation coefficients (R) values between RMSD and environment-specific score for 4state_reduced (Park and Levitt, 1996) from Decoy 'R'Us

PDB ID	Description	Decoy number	Residue number	RMSD range	R
1fc2	Human Fc Fragment	501	43	3.1-10.3	0.411
1hdd-C	Engrailed Homeodomain	501	57	2.8-12.9	0.315
2cro	Phage 434 Cro Protein	501	65	4.3-12.6	0.320
4icb	Calbindin-binding Protein	501	76	4.8-14.1	0.191

Table 6.4 The correlation coefficients (R) values between RMSD and environment-specific score for Fisa (Simons et al., 1997) from Decoy 'R'Us

PDB ID	Description	Decoy number	Residue number	RMSD range	R
1bg8-A	E.coli Hde A	1201	76	6.0-15.8	0.133
1bl0	DNA binding motif in MarA	972	99	3.6-18.2	0.353
1eh2	Eps15 Homology domain	2414	79	4.0-15.3	0.334
1jwe	E.coli Dnab Helicase	1408	114	7.8-20.9	0.245
L30	Unknown	1400	104	6.5-24.6	0.216
Smd3	D3B subcomplex of the human core Snrnp domain	1201	71	8.5-17.0	0.104

Table 6.5 The correlation coefficients (R) values between RMSD and environment-specific score for Fisa casp3 (Simons et al., 1997) from Decoy 'R'Us

PDB ID	Description	Decoy number	Residue number	RMSD range	R
1ash	Ascaris hemoglobin domain I	30	147	2.2-7.0	0.479
1bab-B	Hemoglobin Thionville	30	146	0.7-6.9	0.787
1col-A	Core-forming domain of colicin A	30	197	12.3-30.2	0.523
1cpc-A	C-phycocyanin from Fremyella diplosiphon	30	162	6.8-14.0	0.266
1ecd	Erythrocruorin	30	136	1.5-6.2	0.782
1emy	Asian elephant cyanometmyoglobin	30	153	0.7-9.3	0.847
1flp	Sulfide-reactive hemoglobin from the clam Lucina pectinata	30	142	1.7-7.2	0.846
1gdm	Leghemoglobin	30	153	2.6-8.4	0.802
1hbg	Glycera dibranchiata hemoglobin	30	147	2.1-6.9	0.781
1hbh-A	Deoxyhemoglobin of the Antarctic fish Pagothenia bernacchii	30	142	1.0-6.3	0.786
1hbh-B	Deoxyhemoglobin of the Antarctic fish Pagothenia bernacchii	30	146	1.0-7.3	0.827
1hda-A	Bovine deoxyhemoglobin	30	141	0.5-5.8	0.790
1hda-B	Bovine deoxyhemoglobin	30	145	0.5-5.6	0.794
1hlb	Hemoglobin from Caudina arenicola	30	157	2.9-7.0	0.626
1hlm	Hemoglobin from Caudina arenicola	30	158	3.0-8.7	0.398
1hsy	Myoglobin H64T Mutant	30	153	0.8-9.7	0.909
1ith-A	Hemoglobin from Urechis caupo	30	141	1.6-6.1	0.857
1lht	Myoglobin from Loggerhead Sea Turtle	30	153	0.8-9.7	0.562
1mba	Aplysia limacina myglobin	30	146	1.8-7.3	0.787
1mbs	Seal myoglobin	30	153	1.7-9.3	0.757
1myg-A	Pig metmyoglobin	30	153	0.5-9.6	0.859
1myj-A	Aquomet Myoglobin	30	153	0.6-7.9	0.806
1myt	Myoglobin from Yellow Tuna	30	146	1.0-10.0	0.681
2dhb-A	Horse deoxyhemoglobin	30	141	0.6-6.4	0.817
2dhb-B	Horse deoxyhemoglobin	30	146	0.9-7.1	0.794
2lhb	Lamprey-hemoglobin from Petromyzon marinus	30	149	3.0-8.1	0.606
2pgh-A	Aquomet porcine hemoglobin	30	141	0.7-6.5	0.687
2pgh-B	Aquomet porcine hemoglobin	30	146	0.8-7.5	0.804

4sdh-A	Deoxy hemoglobin I averages	30	145	2.3-6.4	0.663

Table 6.6 The correlation coefficients (R) values between RMSD and environment-specific score for Hg_structal (Samudrala et al., 1998) from Decoy 'R'Us.

PDB ID	Description	Decoy number	Residue number	RMSD range	R
1beo	Beta-cryptogein	2001	95	7.0-15.6	0.115
1ctf	L7/L12 50 S ribosomal protein	2001	68	5.5-12.8	0.035
1dkt-A	Type 1 human cyclin-dependent kinase subunit	2001	72	6.7-14.1	0.029
1fca	Ferredoxin from clostridium Acidurici	2001	55	5.1-11.4	0.036
1nkl	Nk-lysin from Pig	2001	78	5.3-13.6	0.044
1pgb	Protein G (B1 IgG-binding domain)	2001	56	5.8-12.9	0.035
1trl-A	Thermolysin fragment	2001	62	5.4-12.5	0.022
4icb	Calbindin-binding Protein	2001	76	4.7-12.9	0.036

Table 6.7 The correlation coefficients (R) values between RMSD and environment-specific score for Lattice ssfit (Samudrala et al., 1999; Xia et al., 2000) from Decoy 'R'Us.

PDB ID	Description	Decoy number	Residue number	RMSD range	R
1b0n-B	Sinr protein/Sini protein complex	498	31	2.45-6.03	0.352
1bba	Bovine pancreatic polypeptide	501	36	2.78-8.91	0.279
1ctf	L7/L12 50 S ribosomal protein	498	68	3.59-12.5	0.172
1dtk	Dendrotoxin K	216	57	4.32-12.6	0.149
1fc2	Immunoglobulin Fc and fragment B of protein A complex	501	43	3.99-8.45	0.224
1igd	Protein G	501	61	3.11-12.6	0.165
1shf-A	Fyn proto-oncogene tyrosine kinase	438	59	4.39-12.3	0.055
2cro	434 cro protein	501	65	3.87-13.5	0.146
2ovo	Ovomucoid third domain	348	56	4.38-13.4	0.258
4pti	Trypsin inhibitor	344	58	4.94-13.2	0.244

Table 6.8 The correlation coefficients (R) values between RMSD and environment-specific score for lmds (Samudrala and Levitt, 2000) from Decoy 'R'Us.

PDB ID	Description	Decoy number	Residue number	RMSD range	R
1ctf	L7/L12 50 S ribosomal protein	11402	68	4.44-13.0	0.051
1e68	Bacteriocin As-48	11362	70	2.98-12.5	0.108
1eh2	Eps15 Homology domain	11442	95	5.32-15.1	0.054
1khm	Hnrnp K (Kh3)	21081	73	3.46-14.6	0.059
1nkl	Nk-lysin from Pig	11662	78	3.84-14.2	0.060
1pgb	Protein G (B1 IgG-binding domain)	11282	56	4.67-13.0	0.120

Table 6.9 The correlation coefficients (R) values between RMSD and environment-specific score for lsemfold (Samudrala and Levitt, 2002) from Decoy 'R'Us.

The correlation coefficient values of the 4state_reduced decoy set from those of atomic energy functions developed by Gatchell et al. (2000), Lu and Skolnick (2001), Zhou and Zhou (2002), the NN model of TUNE proposed by Lin et al. (2002) and TES are given in Table 6.10. The values in the Table are the correlation coefficients. It shows that the TES has a similar performance pattern with the NN model of TUNE than the other three atomic energy functions. The TES performs better than TUNE on the decoy set 1ctf, 1sn3 and 4pti, worse on 1r69, 2cro, 3icb and 4rxn. Compared with those three atomic energy functions, the TES only performs better on the decoy set 4rxn with KBP and GDV and on decoy set 1sn3 with DFIRE-A. KBP is a heavy atom distance-dependent knowledge-based pairwise potential. DFIRE-A is a residue-specific all-atom potential of mean force. GDV is an atomic energy function that combines molecular mechanics with empirical solvation and entropic terms. Both TES and TUNE are built on less detailed residue level description than KBP, GDV and DFIRE-A. Lu and Skolnick (2001) showed that the details of the potential construction are very important for building energy functions. The threading methods with atom level structure environmental descriptions are more accurate than those with residue level descriptions. That is why TES and TUNE perform worse on most of the decoy sets than KBP, GDV and DFIRE-A. However, due to the

NN model used in TES and TUNE, more information is extracted than normal residue level models. The TES and TUNE show comparable performance with KBP, GDV and DFIRE-A.

Method / PDB code	DFIRE-A[a]	KBP[b]	GDV[c]	TUNE[d]	**TES[e]**
1ctf	0.70	0.667	0.674	0.610	**0.623**
1r69	0.68	0.675	0.641	0.642	**0.555**
1sn3	0.32	0.463	0.524	0.354	**0.419**
2cro	0.75	0.617	0.549	0.625	**0.437**
3icb	0.83	0.829	0.769	0.771	**0.685**
4pti	0.45	0.462	0.473	0.432	**0.443**
4rxn	0.66	0.579	0.582	0.596	**0.586**

Table 6.10 Evaluation of proposed model TES with TUNE, GDV and KBP on 4state_reduced decoy set from Decoy'R'Us

[a] DFIRE-A is the mean-force atomic potential from Zhou and Zhou (2002).

[b] KBP is the atomic potential from Lu and Skolnick (2001).

[c] GDV is the atomic potential developed by Gatchell, et al. (2000).

[d] TUNE is NN model from Lin et al. (2002).

[e] The proposed model in this research.

For the seven decoy sets, the proposed model does not always give the highest score for the native model. Some decoy structures can have a higher score. The TUNE model (Lin et al., 2002) also suffers the same problem. The reason for the failure is not entirely clear. However, there are two studies related to the problem: (1) Decoys are deliberately designed protein sets contain conformations close (<4Å) to the native structure. The TES model is built on the basis of structural information and does not measure the free energy between the interactions of residues. It is a difficult task for the TES model to recognize those decoys which have stabilised structures but not in the native structure energy basin. (2) The TES model is built on the database of SCOP. 535,525 residues are selected for training the TES model. Vendruscolo et al. (2000) mentioned

that for large enough databases, pairwise contact potentials could not stabilise all native folds equally well.

Figure 6.2 shows the correlation between the RMSD and the environment-specific score for the 4state_reduced decoy set. It indicates that the closer the decoy structure is to the native structure, the higher its score is. For example, the last one in Figure 6.2 with the PDB code of 4rxn, the environment-specific score of native protein (with 0Å RMSD) is 38.4, which is the largest value among all those decoys. Most of the near native decoys (RMSD <4Å) have larger environment specific scores than those of non-near native decoys (RMSD >4Å). The Figure demonstrates that TES model has good performance on 4rxn decoy set.

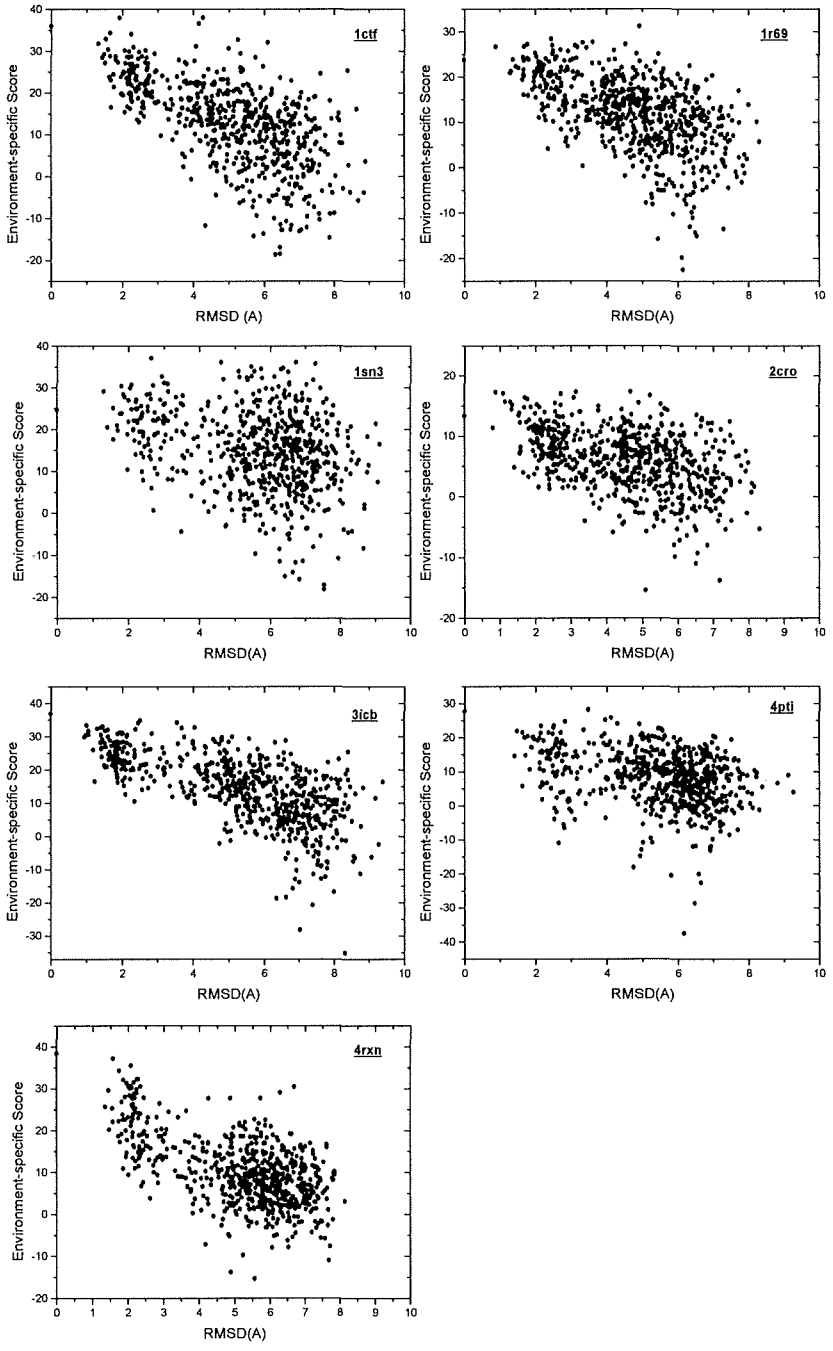


Figure 6.2. RMSD (root-mean-square-deviation) vs. the environment-specific score from the proposed ANN model of seven decoy sets from Park and Levitt (1996).

*6.3.3 ROC curve**

To compare with the other published work, the correlation coefficient is used to evaluate the quality of the score. An alternative way to evaluate the performance of the TES model is using the ROC (Receiver Operating Characteristics) curve (Baldi, et al., 2000). Each conformation of the decoy protein is classified as either "native-like" (RMSD less than 4Å) or "non-native-like" (RMSD greater than 4Å). In this research, given a threshold value, the TP (true positive), TN (true negative), FP (false positive) and FN (false negative) are defined as:

TP = the number of proteins when TES score is larger than the threshold and the decoy protein is native-like;

TN = the number of proteins when TES score is less than the threshold and the decoy protein is nonnative-like;

FP = the number of proteins when TES score is larger than the threshold and the decoy protein is nonnative-like;

FN = the number of proteins when TES score is less than the threshold and the decoy protein is native-like.

The sensitivity ($\frac{TP}{TP+FN}$) versus the false positive rate ($\frac{FP}{FP+TN}$) is plotted as an ROC curve. The performance of two-class prediction is measured by the area under a ROC curve (Hanley and McNeil, 1982). The ROC curve for a perfect prediction model shows no trade-off between

* This part of work is done according to the feedback of submitted paper (Jiang et al., 2005c). ROC map is said to be more proper evaluation methods than correlation coefficient. It is becoming to be standard in bioinformatics field. Unfortunately, till now, no other published evaluation results could be found by ROC map for these benchmark problems applied in this research.

sensitivity and false positive rate, so the value of its area is 1.0. On the contrary, for a random prediction model, the ROC curve is a diagonal from (0, 1) to (1, 0) with the area of 0.5. So the useful range of ROC curve areas is 0.5~1.

For the seven decoy sets from Decoys 'R' Us, the ROC curve is shown in Figure 6.3. The values of ROC area are shown in Table 6.11. They are ranged from 0.69 (4pti) to 0.89 (3icb). Since the ROC value of random prediction model is 0.5, it showed that the overall performance of TES model is considerably greater than random prediction model would produce.

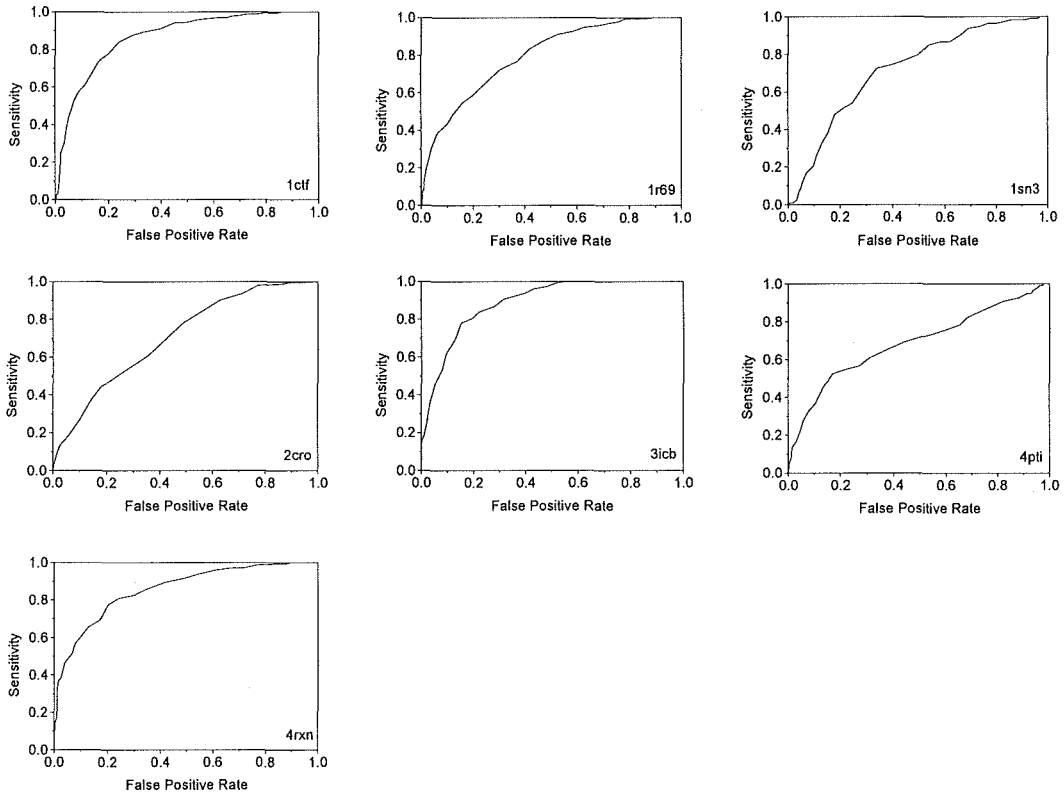


Figure 6.3. False positive rate-sensitivity plots for TES model.

PDB code	1ctf	1r69	1sn3	2cro	3icb	4pti	4rxn
ROC value	0.863	0.791	0.715	0.705	0.886	0.686	0.856

Table 6.11 The ROC value of Decoys'R'Us, which is measured by the area under the ROC curve.

6.4 Discussion

In this Chapter, the design and evaluation of a threading score following the contact potential approach is described. The performance of the TES score is evaluated with two benchmarks on the discrimination of protein native structure and decoys. The experimental results showed that the TES model with residue environmental description is compatible to those potential energy functions with the detailed atomic level structural environment description. It has also been demonstrated that the TES can outperform those of residue level contact potentials and the TUNE model. The threading scores derived from the TES model are log-odds, which are similar to widely applied amino acid substitution matrices such as BLOSUM62 (Henikoff and Henikoff, 1992). These residue-specific log-odds can be employed by protein alignment algorithms, such as double dynamic programming (Jones et al., 1992) and a divide-and-conquer algorithm (Xu and Xu, 2000), to built a threading program.

In the following final Chapter of this thesis, the conclusions of the research work will be given. The possible future work of this thesis will be delineated.

CHAPTER 7 SUMMARY AND FUTURE WORK

7.1 Summary

This thesis has demonstrated that the machine learning approach (NNs and SVMs) is an effective way for solving the protein threading problem. An efficient and effective framework for protein threading is developed and its performance is validated. The results have shown that the machine learning approach helps to increase the prediction accuracy while potentially significantly decreasing the computational load. Compared with one of the better performing threading model PROSPECT, which took about 45 hours to predict one target (t0174) in CASP4, the MESSM can make an alignment for sequence with 150 amino acids in 30 seconds. The threading model developed in this research may be considered as an alternative tool for protein prediction. Part of the research work has been published in both conferences and journals (Jiang et al., 2005a; 2005b; 2005c; 2004).

7.1.1 Achievement of the work

The success of the developed threading model is due to four key factors.

Firstly, as NNs are suitable tools for finding statistical correlations, they are used to map the amino acid substitution relations in the framework of MESSM. With the more precise structural information extracted by the NN, the substitution probability of each pair of amino acids at any chosen structural environment can be generated. Furthermore, the NN is used in training the sequence-structure compatibility for our TES threading score. The BPNN is adopted with three layers. Different architectures of the NN are tested by changing the neurons in the hidden layer. The best performance NN is chosen to map the amino acid substitution and the sequence-structure compatibility. Unlike a normal computationally intensive energy potential process, the inclusion of the BPNN within the threading model appears to help increase accuracy while potentially significantly decreasing the computational load.

Secondly, by combining the environment-specific information with the sequence-specific information, a mixed substitution score is built by the inclusion of the structurally-derived substitution mapping and the well-developed amino acid substitution matrix BLOSUM30. We have conducted some experiments on comparing the performances of models with the mixed substitution score and structure or sequence information alone. By optimizing a combined parameter, the experimental results demonstrate that the protein threading model with the mixed substitution mapping has a better performance than the one with either structure or sequence information only. The positive consensus combination allows this method to exhibit comparable results to threading models under various cases.

Thirdly, in contrast to the traditional expert human interpretation on recognizing the best fit templates, the SVM as a new generation of machine learning algorithm is adopted to select the best templates for each target sequence. The experimental results (specificity-sensitivity curves) indicate that the SVM can help to derive a significant high reliable score

function (higher sensitivity than other models on the same specificity) for the template selection. Thus, comparable fold recognition results are achieved to those of models with expert human interpretation of results. Using a SVM in the MESSM framework makes it to be an automated model to suit the fast genome sequencing.

Finally, with the extended research work on building a threading score TES, the results demonstrated that the residue contact measuring scheme is a simple and efficient measurement compared to most other threading programs. For each residue pair, only the two distances (side-chain to side-chain and side-chain to main-chain) are considered for computing. It helps to save a lot computational cost compared to those scoring functions with atom level structure environment description. The performance of the TES score is comparable to current potential energy models with detailed atomic level structure environment description, such as KBP (Lu and Skolnick, 2001), GDV (Gatchell, et al., 2000) and DFIRE-A (Zhou and Zhou, 2002). It outperformed those of residue contact potentials (for example, RKBP (Skolnick et al., 2000) and CDF (Samudrala and Moult, 1998)) and TUNE model (Lin et al., 2002) which is also a NN based threading score but with a different residue contact measurement.

7.1.2 Discussion

Besides the four factors that contribute to the success of our threading model, two components are also important in determining the power of the developed threading model. They are the number of genuinely diverse sequences within a superfamily and the accuracy of the structural alignment. The first component is addressed by the efforts of many researchers to elucidate protein structures using physical techniques. The structure database is growing rapidly, and consequently so are the superfamilies. As the developed model is based on machine learning, it is

easy to be re-trained with new data. The threading framework can be easily updated with the growth of the structure database. However, the accuracy of the structural alignment poses many challenges. The concept of an "accurate structural alignment" is not clearly defined. For a given pair of structurally homologous proteins, there is rarely an unambiguously correct alignment. Even when there is such a rare case, automatic determination of the alignment is far from the experimental result. FLASH (Shih and Hwang, 2003) is adopted in this research. FLASH was argued to be one of the best performances structural alignment programs when this research was carried on, although there are several equally powerful programs available for structural alignment (e.g. SAP (Orengo et al., 1992), DALI (Holm and Sander, 1998), VAST (Madej et al., 1995)). Instead of providing perfect alignments (and now there is an absence of an agreed definition of perfect), these protein structural alignment programs currently can only try to find an alignment that is close in quality to an expert's manual alignment. With the development of structural alignment programs, the performance of our designed framework would be expected to be improved as well.

This thesis is concentrated on the improvement of efficiency while retain the accuracy of prediction. There is a trade-off between the computational cost reduced in the threading framework and the required prediction accuracy. The threading methods with atom level structure environmental descriptions are likely to improve the effectiveness but require a higher computational cost. Due to the machine learning approach, the MESSM framework achieves a comparable performance on protein prediction, though MESSM is built with residue level structure environmental description. However, with more structural and sequence evolutionary information to be imported into current MESSM framework, as described in future work of Section 7.2, an improvement on effectiveness is expected.

7.1.3 Limitation of the work

This thesis has described the development of a machine learning based threading framework MESSM. MESSM is the first version of the threading model. It can achieve only comparable fold recognition results with current threading models based on energy potentials. However, further improvement could be applied to our current MESSM model, as described in section 7.2. A better fold recognition performance is expected.

Within contact potential approach, a threading score (TES) instead of a full threading program is developed. It can only be used to discriminate the protein native and decoys. Hopefully, in the future, a full threading program with heuristic algorithms will be implemented.

7.2 Future work

The novel machine learning approach to protein threading opens a new process to develop models in protein structure prediction. It has room to be further developed.

The current MESSM framework reduced the computational load and gained a comparable performance on prediction accuracy. Thus, in the future, the main improvement of MESSM framework is concentrated on the increasing of the effectiveness.

Residue contact measurement (increase prediction accuracy) The residue contact measuring scheme is a key factor to affect the performance of protein threading. The residue contact measurement of the current MESSM model depends on the distance. It shows some improvement compared to the previous work. However, there might exist such a case

that even the distance between two residues is larger than a water molecule, if there does not exist a third residue or water molecule sitting inside these two residues, they will has some distant contact, although in most of the case, these contact could be ignored. Considering such a situation, a three dimensional measurement could do a better job than the current one. 3D Voronoi Diagrams might be one of the choices.

Given n points in space, Voronoi Diagrams divide the space into n regions such that each region contains exactly one point (generating point) and every point in the given region is closer to the generating point than to any other. Thus, suppose given each residue main chain center (α carbon) and side chain center in a protein three-dimensional structure space, the space could be partitioned into regions using Voronoi diagrams. Each region is generated by and contains one residue main chain center (α carbon) or side chain center. The residue contact only happens for each pair of neighbour regions when the two regions are not generated by the same residue. In this way, the residue contact could be counted accurately. A threading model based on this residue contact measuring scheme could get its performance improved.

Secondary structure component (increase prediction accuracy) Secondary structure element alignment, using observed and predicted secondary structure, have previously been incorporated with protein threading (Kelley et al., 2000; Shi et al., 2001). Recently some studies have been carried on how the incorporation of secondary structure component can improve the fold recognition performance of the threading model (McGuffin and Jones, 2003). Since structure is better conserved than the sequence between distantly related proteins, the incorporation of such structure information could therefore benefit the accuracy of protein fold recognition. Thus, by considering secondary structure into the current framework of MESSM, a further improvement on MESSM can be expected.

Currently, there are several predictive methods available for protein secondary structure prediction, such as PSI-PRED (McGuffin et al., 2000), PHD (Rost et al., 1994) and DSC (King et al., 1997). The secondary structure information could be implemented for both the query and template proteins. A profile should be built for query sequence including the information of the predicted secondary structure and the corresponding sequence. The observed and predicted secondary structure information for protein templates could be added as a component of the scoring function. The sequence-profile alignment used in MESSM needs to be replaced by profile-profile alignment.

Multiple alignments (increase prediction accuracy) In multiple alignments, protein sequences are aligned optimally by bringing the greatest number of similar characters into regions. Such regions may represent conserved functional or structural domains. It is generally agreed that information from multiple alignments can help to refine a pairwise alignment of sequences. During the last few years, it has been shown that the methods with the inclusion of multiple alignments are superior to methods using single sequence only (Wallner et al., 2004). Therefore, the MESSM program is expected to be improved by inclusion of multiple alignment information.

There are two kinds of multiple alignment information for proteins. They are structural alignments and sequence alignments. The structural alignment of more proteins could provide more precise information than sequence alignments, but it is only possible when the three-dimensional structures of all the proteins to be aligned are known. Currently there are not enough known proteins in the database for structural alignments. Therefore, only multiple sequence alignments can be adopted for both the query and template proteins. There are several methods available for protein multiple sequence alignments, such as T-Coffee (Notredame et al., 2000), ClustalW (Thompson et al., 1994) and MUSCLE (Edgar, 2004). Like

secondary structure component, the multiple alignment information for templates could be built as one of the components of scoring function. The multiple alignment information for query sequence needs to be added into a query profile. Several profile-profile alignment methods (e.g. Marti-Renom et al., 2004; Yona and Levitt, 2002) could be implemented.

Threading program on contact potential approach (build a functional threading model) In this research, a TES model is built following contact potential approach. The TES model is not a threading program. However, the environment-specific scores from the TES method are log-odds. They can be employed for protein alignment algorithms to build a threading program. In the future, a heuristic algorithm, such as double dynamic programming (Jones et al., 1992) and a divide-and-conquer algorithm (Xu and Xu, 2000) are expected to be employed into TES model in order to develop a full threading program for protein prediction.

Accuracy and efficiency study In this research, the main focus is to retain accuracy of prediction against the reduction of computation time involved in the protein threading. By doing so, a residue level environment description is used in the framework of MESSM though models with atom level environment description have been proved to be more accurate. In the future, a study will be carried out on the trade-off between the accuracy and efficiency. Another framework will be built on the atom level environment description. The computational time versus prediction accuracy between two frameworks will be analysed and discussed. A user menu for selection between the two frameworks will be built. For those applications require a fast but not as accurate as possible answer, the current MESSM could be used. For those applications require more accurate answer but not caring about computational time, the new framework can be applied.

REFERENCE

Altschul, S. and Erickson, B. W. (1986). Optimal sequence alignment using affine gap costs. *Bulletin of Mathematical Biology,* 48:603 -616.

Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D. J. (1997) Gapped BLAST and PSI_BLAST: a new generation of protein database search programs. *Nucleic Acids Research,* **25**, 3389-3402.

Anderson, D. C., Li, W., Payan, D. G. and Noble, W. S. (2003) A new algorithm for the evaluation of shortgun peptide sequencing in proteomics: SVM classification of peptide MS/MS spectra and SEQUEST scores. *Journal of Proteome Research,* 2, 137-146.

Anfinsen, C. B. (1973) Principles that govern the folding of protein chains. *Science,* **181**(96), 223-30.

Bairoch, A. and Apweiler, R. (1996) The SWISS-PROT protein sequence data bank and its new supplement TREMBL. *Nucleic Acids Research,* **24**(1), 21-25.

Bairoch, A. and Apweiler, R. (2000) The SWISS-PROT protein sequence data bank and its new supplement TREMBL in 2000. *Nucleic Acids Research,* **28,** 45-48. <http://www.expasy.ch/sprot>

Baker, D. and Sali, A. (2001) Protein structure prediction and structural genomics. *Science,* **294**(5540), 93-96.

Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. A. F. and Nielsen, H. (2000) Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics,* **16**(5), 412-424.

Baldi, P. and Brunak, S. (2001) *Bioinformaics: The Machine Learning Approach,* MIT Press, Cambridge, **MA.**

Bates, P. A., Kelley, L. A., MacCallum, R. M. and Sternberg, M. J. (2001) Enhancement of protein modelling by human intervention in applying the automatic programs 3D-JIGSAW and 3D-PSSM. *Proteins: structure, function, and bioinformatics,* **45**(Suppl 5), 39-46.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N. and Weissig, H. et al. (2000) The Protein Data Bank. *Nucleic Acids Research,* **28**, 235-242.

Bernstein, F. C., Koetzle, T. F., Williams, G. J. B., Meyer Jr, E., Brice, M. D. Rodgers, J. R., Kennard, O., Shimanouchi, T. and Tasumi, M. (1977) The Protein Data Bank: a computer-based archival file for macromolecular structures. *Journal of Molecular Biology,* **112(3),** 535-542.

Benson, D. A., Karsch, M. I., Lipman, D. J., Ostell, J., Rapp, B. A. and Wheeler, D. L. (2000) GenBank. *Nucleic Acids Research,* **28**, 15-18. <http://www.ncbi.nlm.nih.gov/>

Blake, J. D. and Cohen, F. E. (2001) Pairwise sequence alignment below the twilight zone. *Journal of Molecular Biology,* 307: 721-735.

Bohr, H., Bohr, J., Brunak, S., Cotterill, R. M. J. and Fredholm, H. et al. (1990) A novel approach to prediction of the 3-dimensional structures of protein backbones by NNs. *FEBS Letters,* **261**, 43-46.

Bohr, J., Bohr, H., Brunak, S., Cotterill, R. M. J., Fredholm, H., Lautrup, B. and Petersen, S. B. (1993) Protein structures from distance inequalities. *Journal of Molecular Biology,* **231**, 861–869.

Bonneau, R., Tasi, J., Ruczinski, I., Chivian, D., Rohl, C., Strauss, C. E. and Baker, D. (2001) Rosetta in CASP4: Progress in ab initio protein structure prediction. *Proteins: structure, function, and bioinformatics,* **45**(Suppl 5), 119-126.

Bork, P. (1991) Shuffled domains in extracellular proteins. *FEBS Lett,* **286**(1-2), 47-54.

Boser, B. E., Guyon, I. M. and Vapnik, V. (1992) A training algorithm for optimal margin classifiers. *Proceeding of the 5th Annual ACM Workshop on Computational Learning Theory*, 144-152.

Bowie, J. U., Luthy, R. and Eisenberg, D. (1991) A method to identify protein sequences that fold into a known 3-dimensional structure. *Science*, **253**(5016), 164-170.

Brenner, S. E., Chothia, C., Hubbard, T. J. P. (1998) Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. *Proceedings of National Academy of Science of the United States of America*, **95**, 6073-6078.

Brooks, C. L., Karplus, M. and Pettitt, B. M. (1990) *Proteins: A theoretical perspective of dynamic, structure and thermodynamics.* John Wiley and Sons, New York.

Brown, M.P.S., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C.W., Furey, T.S., Ares, M. and Haussler, D (2000) Knowledge-based analysis of microarray gene expression data by using Support Vector Machines. *Proceedings of National Academy of Science of the United States of America*, 97:262-267.

Brunak, S., Engelbrecht, J. and Knudsen, S. (1991) Prediction of human mRNA donor and acceptor sites from the DNA sequence. *Journal of Molecular Biology*, **220**, 49-65.

Bryant, S. H. and Lawrence, C. E. (1993) An empirical energy function for threading protein sequence through folding motif. *Proteins: structure, function, and genetics*, **16**(1), 92-112.

Bryant, S. H. and Altschul, S. F. (1995) Statistics of sequence-structure threading. *Current Opinion in Structural Biol.*, **5**, 236-244.

Bryant, S. H. (1996) Evaluation of threading specificity and accuracy. *Proteins: structure, function, and bioinformatics*, **26**(2), 172-185.

Bult, C. J., White, O., Olsen, G. J., Zhou, L., Fleischmann, R. D., Sutton, G. G., Blake, J. A., FitzGerald, L. M., Clayton, R. A., Gocayne, J. D., Kerlavage, A. R., Dougherty, B. A., et al. (1996) Complete genome

sequence of the methanogenic archaeon, Methanococcus jannaschii. Science, 273, 1058-1073.

Burges, C. J. (1998) A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery.* 2(2), 121-167.

Cai, Y. D., Liu, X. J., Xu, X. B. and Zhou, G. P. (2001) Support vector machines for predicting protein structural class. *BMC Bioinformatics,* 2:3.

Carter, R. J., Dubchak, I. And Holbrook, S. R. (2001) A computational approach to identify genes for functional RNAs in genomic sequences. *Nucleic Acids Research,* 29(19):3928-3938.

Chandonia, J. M. and Karplus, M. (1995) Neural networks for secondary structure and structural class predictions. *Protein Science,* **4**, 275-285.

Chothia, C. and Lesk, A. (1986) The relation between the divergence of sequence and structure in proteins. *EMBO Journal,* **5**(4), 823-826.

Collins, F. S., Morgan, M. and Patrinos, A. (2003) The Human Genome Project: Lessons from Large-Scale Biology, *Science,* 11, 286.

Compiani, M., Fariselli, P., Martelli, P. L. and Casadio, R. (1998) An entropy criterion to detect minimally frustrated intermediates in native proteins. *Proceedings of National Academy of Science of the United States of America,* **95**(16), 9290-9294.

Cristianini, N. and Shawe-Taylor, J (2000) *An introduction to Support Vector Machines and other kernel-based learning methods.* Cambridge University Press.

Cybenko, G. (1989) Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signal, and Systems,* **2**(4): 303-314.

Dayhoff, M.O., Schwartz, R.M., and Orcutt, B.C. (1978) A model for evolutionary change. *Atlas of protein sequence and structure* (ed. Dayhoff, M.O.), 5 (S3), 345-352. National Biomedical Research Foundation, Washington, D.C.

Degroeve, S., De Baets, B., Van de Peer, Y. and Rouz, P. (2002) Feature subset selection for splice site prediction. *Bioinformatics,* 18:S75-S83.

Ding, C. and Dubchak, I. (2001) Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, 17:349-358.

Domingues, F. S., Lackner, P., Andreeva, A. and Sippl, M. J. (2000) Structure-based evaluation of sequence comparison and fold recognition alignment accuracy. *Journal of Molecular Biology*, **297**:1003-1013.

Eddy, S. R. (1998) Profile hidden Markov models. *Bioinformatics*, 14:755-763.

Edgar, R. C. (2004), MUSCLE: multiple sequence alignment with high accuracy and high throughput, *Nucleic Acids Research*, **32**(5), 1792-97.

Elofsson, A., Fischer, D., Rice, D. W., LeGrand, S. M. and Eisenberg, D. (1996) A study of combined structure-sequence profiles. *Folding & Design*, **1**, 451-461.

England, J. L., Shakhnovich, B. E. and Shakhnovich, E. I. (2003) Nature selection of more designable folds: A mechanismfor thermophilic adaptation. *Proceedings of the National Academy of Sciences of the U.S.A.*,**100**, 8727-8731.

Fariselli, P. and Casadio, R. (1999) A neural network based predictor of residue contacts in proteins. *Protein Engineering*, **12**(1), 15-21.

Fariselli, P., Olmea, O., Valencia, A., Casadio, R. (2001) Prediction of contact maps with neural networks and correlated mutations. *Protein Engineering*, **269**, 835-843.

Felts, A.K., Gallicchio, E., Wallqvist, A., and Levy, R.M. (2002) Distinquishing native conformations of proteins from decoys with an effective free energy estimator based on the OPLS all-atom force field and the surface generalized Born solvent model. *Proteins: structure, function, and bioinformatics*, **48:** 404–422.

Fischer, D. and Eisenberg, D. (1996) Protein Fold Recognition Using Sequence-Derived Properties. *Protein Science*, 5: 947-955.

Fischer, D., Elofsson, A., Rice, D. and Eisenberg, D (1996) Assessing the performance of fold recognition methods by means of a comprehensive benchmark. *Proceedings of the Pacific Symposium on Biocomputing,* 300-318.

Fischer, D., Rychlewski, L., Dunbrack Jr., R.L., Ortiz, A.R., and Elofsson, A. (2003) CAFASP3: The third critical assessment of fully automated structure prediction methods. *Proteins: structure, function, and bioinformatics,* **53** (Suppl 6): 503–516.

Fleischmann, R. D., Adams, M. D., White, O., Clayton, R. A., Kirkness, E. F., Kerlavage, A. R., Bult, C. J., Tomb, J. F., Dougherty, B. A., Merrick, J., M., et al. (1995) Whole-genome random sequencing and assembly of Haemophilus influenza Rd. Science, 269,496-512.

Flockner, H., Braxenthaler, M., Lackner, P., Jaritz, M., Ortner, M. and Sippl, M. J. (1995) Progress in fold recognition. *Proteins: structure, function, and genetics,* **23**, 376-386.

Furey, T.S., Cristianini, N., Duffy, N. Bednarski, D. W., Schummer, M. and Hussler, D. (2000) Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics,* 16, 906-914.

Gatchell, D. W., Dennis, S. and Vajda, S. (2000) Discrimination of nearnative protein structures from misfold models by empirical free energy functions. *Proteins: structure, function, and bioinformatics,* **41**, 518-534.

Gerstein, M. (1998) Patterns of protein-fold usage in eight microbial genomes: a comprehensive structural census. *Proteins: structure, function, and bioinformatics,* **33**, 518-534.

Gibas, C. and Jambeck, P. (2001) *Developing Bioinformatics Computer Skills,* O'Reilly & Associates Inc.

Godzik, A., Skolnick, J. and Kolinski, A. (1992) Topology fingerprint approach to the inverse folding problem. *Journal of Molecular Biology,* **227**, 227-238.

Gonnet, G. H., Cohen, M. A. and Benner, S. A. (1992) Exhaustive matching of the entire protein sequence database. *Science*, **256**, 1433-1445.

Gotoh, O. (1982). An improved algorithm for matching biological sequences. *Journal of Molecular Biology*, **162**, 705-708.

Govindarajan, S., Recabarren, R. and Goldstein, R. K. (1999) Estimating the total number of protein folds. *Proteins: structure, function, and bioinformatics*, **35**, 408-414.

Guyon, I., Weston, J., Barnhill, S. and Vapnik, V. (2002) Gene selection for cancer classification using support vector machines. *Machine Learning*, 46, 389-422.

Hanley, J. A. and McNeil, B. J. (1982) The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, **143**, 29-36.

Hartman, E. J., Keeler, J. D. and Kowalski, J. M. (1990) Layered neural networks with Gaussian hidden units are universal approximations. *Neural Computation*, 2(2), 210-215.

Henikoff, S. and Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proceedings of National Academy of Science of the United States of America*, **89**, 10915-10919.

Henikoff, S. and Henikoff, J. G. (1994) Position-based sequence weights. *Journal of Molecular Biology*, **243**(4), 574-578.

Henikoff, S., and Henikoff, J. G. (1997) Embedding strategies for effective use of information from multiple sequence alignments. *Protein Science*, **6**(3), 698-705.

Hirst, J. D. and Sternberg, M. J. E. (1992) Prediction of structural and functional features of protein and nucleic acid sequences by artificial neural networks. *Biochemistry*, **31**, 7211-7218.

Holley, H. L. and Karplus, M. (1989) Protein secondary structure prediction with a neural network. *Proceedings of the National Academy of Sciences of the U.S.A.*, **86**, 152-156.

Holm, L. and Sander, C. (1992) Evaluation of protein models by atomic solvation preference. *Journal of Molecular Biology,* **225**, 93-105.

Holm, L. and Sander, C. (1997) Dali/ FSSP classification of three-dimensional protein folds. *Nucleic Acids Research,* **25**, 231-234.

Holm, L and Sander, C. (1998) Dictionary of recurrent domains in protein structures. *Proteins: structure, function, and bioinformatics,* **33**(1), 88-96.

Honig, B. (1999) Protein folding: from the levinthal paradox to structure prediction. *Journal of Molecular Biology,* **293**(2), 283-93.

Hua, S. and Sun, Z. (2001a) Support vector machine approach for protein subcellular localization prediction. *Bioinformatics,* 17(8):721-728.

Hua, S. and Sun, Z. (2001b) A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. *Journal of Molecular Biology,* 308, 397-407.

Huang, E., Subbiah, S., and Levitt, M. (1995) Recognizing native folds by the arrangement of hydrophobic and polar residues. *Journal of Molecular Biology,* **252**, 709-720.

Huber, T., Russell, A. J., Ayers, D. and Torda, A. E. (1999) Sausage: protein threading with flexible force fields. *Bioinformatics,* **15**(12), 1064-1065.

Jaakkola, T., Diekhans, M. and Haussler, D. (1999a) Using the Fisher kernel method to detect remote protein homologies. *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology,* AAAI Press, 149-158.

Jaakkola, T., Diekhans, M. and Haussler, D. (1999b) A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology,* 7(1-2):95-114.

Jasny, B. R. and Roberts, L. (2003) Building on the DNA Revolution, *Science,* 11, 277.

Jiang, N., Wu, W. and Mitchell, I (2005a) Protein Fold Recognition Using Neural Networks and Support Vector Machines, *Lecture Notes in Computer Science,* **3578** (IDEAL 2005), 462-469.

Jiang, N., Wu, W. and Mitchell, I (2005b) Protein Fold Recognition by Mixed Environment-Specific Amino Acid Substitution Mapping Using Neural Networks, *Proceeding of the Eleventh ICPADS,* 341-345.

Jiang, N., Wu, W. and Mitchell, I (2005c) Threading with Environment-specific Score by Artificial Neural Networks, *Soft Computing,* Vol. 10(4), 305-314.

Jiang, N., Wu, W. and Mitchell, I (2004) Protein Threading with Residue-environment Matching by Artificial Neural Networks , *Proceedings of the 2004 ACM symposium on applied computing,* **1**, 209-210.

Joachims, T. (1999) Making large-Scale SVM learning practical. *Advances in Kernel Methods - Support Vector Learning,* MIT press.

Johnson, J. L., Rajagopalan, K. V., Mukund, S. and Adams, M. W. W. (1993) Identification of molybdopterin as the organic component of the tungsten cofactor in four enzymes from hyperthermophilic archaea *Journal of Biological Chemistry,* **268**, 4848-4853.

Jones, D. T., Taylor, W. R. and Thornton, J. M. (1992) A new approach to protein fold recognition. *Nature,* **358**, 86-89.

Jones, D. T., Miller, R. T. and Thornton, J. M. (1995) Successful protein fold recognition by optimal sequence threading validated by rigorous blind testing. *Proteins: structure, function, and genetics,* **23**, 387-397.

Jones, D. T. and Thornton, J. M. (1996) Potential energy functions for threading. *Current opinion in Structural Biology,* **6**(2), 210-216.

Jones, D. T. (1999) GenTHREADER: An efficient and Reliable Protein Fold recognition Method for Genomic Sequences. *Journal of Molecular Biology,* **287**, 797-815.

Jones, D.T. and Hadley, C. (2000) Threading methods for protein structure prediction. *Bioinformatics, sequence, structure and databanks* (eds. D. Higgins and W. Taylor), 1-13. Oxford University Press, Oxford, UK

Jones, D. T. (2001) Predicting novel protein folds by using FRAGFOLD. *Proteins: structure, function, and bioinformatics,* Supplement **5**:*127-132.*

Kabsch, W. and Sander, C. (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, **22**(12), 2577-2637.

Karchin, R., Karplus, K. and Haussler, D. (2002) Classifying G-protein coupled receptors with support vector machines. *Bioinformatics*, 18:147-159.

Karplus, K., Karchin, R., Barrett, C., Tu, S., Cline, M., Diekhans, M., Grate L., Casper, J. and Hughey, R. (2001) What is the value added by human intervention in protein structure prediction? *Proteins*, 45(S5), 86-91.

Kanehisa, M. and Bork, P. (2003) Bioinformatics in the post-sequence era. *Nat. Genet.* **33**(Suppl.), 305-310.

Karlin, S., Dembo, A. and Kawabata, T. (1990) Statictical composition of high-scoring segments form molecular sequences. *Annals of Statistics*, **18**, 571-581.

Kelley, L. A., MacCallum, R. M. and Sternberg, M. J. (2000) Enhanced genome annotation using structural profiles in the program 3D-PSSM. *Journal of Molecular Biology*, **299**(2), 499-520.

Kim, D., Xu, D., Guo, J.T., Ellrott, K. and Xu, Y. (2003) PROSPECT II: protein structure prediction program for genome-scale applications. *Protein Engineering*, 16, 641–650.

King, R. D., Saqi, M., Sayle, R. and Sternberg, M. J. (1997) DSC: public domain protein secondary structure prediction. *Comut. Appl. Biosci.*, 13, 473-474.

Kirkpatrick, S., Gelatt Jr. C. D. And Vecchi, M. P. (1983) Optimization by simulated annealing. *Science*, **220**.671-680.

Kneller, D. G., Cohen, F. E. and Langridge, R. (1990) Improvements in protein secondary structure prediction by an enhanced neural network. *Journal of Molecular Biology*, **214**, 171-182.

Koike, A. and Takagi, T. (2004) Prediction of protein–protein interaction sites using support vector machines. *Protein Engineering Design and Selection*, 17(2):165-173.

Kolinski, A., Jaroszewski, L., Rotkiewicz, P. and Skolnick, J. (1998) An efficient Monte Carlo model of protein chains. Modeling the short-range correlations between side group centers, *Journal of Physical Chemistry B*, 102 (23), 4628-4637.

Lackner, P., Koppensteiner, W. A., Domingues, F. S., Sippl, M. J. (1999) Automated large scale evaluation of protein structure predictions. *Proteins: structure, function, and bioinformatics,* **37**(S3), 7-14.

Lapedes, A., Barnes, C., Burks, C., Farber, R. and Sirotkin, K. (1989) Application of neural networks and other machine learning algorithms to DNA sequence analysis. In *Computers and DNA, SFI Studies in the Science of Complexity,* eds G. I. Bell and T. G. Marr, **7**, 157-182, Addison-Wesley, Rosewood City, CA.

Lathrop, R. H. and Smith, T. F. (1996) Global Optimum Protein Threading with Gapped Alignment and Empirical Pair Potentials. *Journal of Molecular Biology,* **255**, 641-665.

Lathrop, R. H., Rogers, R. G. Jr, Smith, T. F. and White, J. V. (1998) A Bayes-optimal sequence-structure theory that unifies protein sequence-structure recognition and alignment. *Bulletin of Mathematical Biology,* 60(6), 1039-1071.

Lazaridis, T. and Karplus, M. (2000) Effective energy functions for protein structure prediction. *Current opinion in Structural Biology,* **10**(2), 139-145.

Lee, M. C. And Duan, Y. (2004) Distinguish protein decoys by using a scoring function based on a new AMBER force field, short molecular dynamics simulations, and the generalized born solvent model. *Proteins: structure, function, and bioinformatics,* **55**(3):620-634.

Lee, Y. and Lee, C. K. (2003) Classification of multiple cancer types by multicategory support vector machines using gene expression data. *Bioinformatics.* 19(9):1132-1139.

Lesk, A. M. (2002) *Introduction to bioinformatics,* Oxford University Press.

Leslie, C., Eskin, E., Weston, J. and Noble, W. S. (2003) Mismatch string kernels for SVM protein classification. *Advances in Neural Information Processing Systems,* MIT Press, Cambridge, MA.

Liao, L. and Noble, W. S. (2002) Combining pairwise sequence similarity and support vector machines for remote protein homology detection. *Proceedings of the Sixth Annual International Conference on Computational Molecular Biology,* 225-232.

Lin, K., May, A. C. W. and Taylor, W. R. (2002) Threading using neural network: the measure of protein sequence-structure compatibility. *Bioinformatics,* **18**(10), 1350-1357.

Lindahl, E. and Elofsson, A. (2000) Identification of Related Proteins on Family, Superfamily and Fold Level. *Journal of Molecular Biology,* **295**, 613-625.

Lo Conte, L., Brenner, S. E., Hubbard, T. J. P., Chothia, C., and Murzin, A. (2002) SCOP database in 2002: refinements accommodate structural genomics. *Nucleic acids research,* **30**(1), 264-267.

Lu, H. and Skolnick, J. (2001) A distance-dependent atomic knowledge-based potential for improved protein structure selection. *Proteins: structure, function, and bioinformatics,* **44**, 223-232.

Lund, O., Frimand, K., Gorodkin, J., Bohr, H., Bohr, J., Hansen J. and Brunak, S. (1997) Protein distance constraints predicted by neural networks and probability density functions. *Protein Engineering,* **10**, 1241-1248.

Luscombe, N. M., Greenbaum, D. and Gerstein, M. (2001) What is bioinformatics? A proposed definition and overview of the field. *Methods Inf Med,* **40**, 346-58.

Machler-Bauer, A. and Bryant, S. H. (1997) A measure of success in fold recognition. *Trends in Biochemical Science,* **22**, 236-240.

Madej, T., Gibrat, J. F. and Bryant, S. H. (1995) Threading a database of protein cores. *Proteins: structure, function, and Genetics,* **23**, 356-369.

Mallick, P., Weiss, R. and Eisenberg, D. (2002) The directional atomic solvation energy: an atom-based potential for the assignment of protein sequences to known folds. *Proceedings of National Academy of Science of the United States of America*, **99**(25):16041-16046.

Marti-Renom, M. A., Madhusudhan, M. S. and Sali, A. (2004) Alignment of protein sequences by their profiles. *Protein Science*, 13, 1071-1087.

McCulloh, W. S. and Pitts, W. (1943) A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*, Vol. 5, 115-133.

McGuffin, L. J., Bryson, K. and Jones, D. T. (2000) The PSIPRED protein structure prediction server. *Bioinformatics*. 16, 404-405.

McGuffin, L. J. and Jones D. T. (2003) Improvement of the GenTHREADER method for genomic fold recognition. *Bioinformatics*, **19**, 874-881.

Milik, M., Kolinski, A. and Skolnick, J. (1995) Neural network system for the evaluation of side-chain packing in protein structures. *Protein Engineering*, **8**, 225-236.

Mizuguchi, K., Deane, C. M., Blundell, T. L. And Overington, J. P. (1998) HOMSTRAD: a database of protein structure alignments for homologous families. *Protein Science*, **7**, 2469-2471.

Moler, E. J., Chow, M. L. and Mian, I. S. (2000) Analysis of molecular profile data using generative and discriminative methods. *Physiol Genomics*, 4:109-126.

Mosimann, S., Meleshko, R. and james, M. (1995) A critical assessment of comparative molecular modelling of tertiary structures in proteins. *Proteins: structure, function, and genetics*, **23**, 301-317.

Moult, J., Judson, R., Fidelis, K. and Pedersen, J. T. (1995) A large-scale experiment to assess protein structure prediction methods. *Proteins: structure, function, and genetics*, **23**, □-□.

Moult, J. (1997) Comparison of database potentials and molecular mechanics force fields. *Current opinion in Structural Biology*, **7**(2), 194-199.

Mukherjee, S., Tamayo, P., Slonim, D., Verri, A., Golub, T., Mesirov, J. and Poggio, T. (1999) Support vector machine classification of microarray data. *Technical Report AI Memo 1677,* Massachusetts Institute of Technology.

Mukherjee, S. and Vapnik, V. (1999) Multivatiate density estimation: an svm approach. *Technical Report AI Memo 1653,* Massachusetts Institute of Technology.

Murzin, A. G., Brenner, S. E., Hubbard, T. and Chothia, C. (1995) SCOP: a structural classification of proteins database for the invertigation of sequences and structures. *Journal of Molecular Biology,* **241**(4), 536-540.

Myers, E. W. and Miller, W. (1988). Optimal alignments in linear-space. *Comput. Appl. Biosci.* **4**, 11-17.

Needleman, S. B. and Wunsch, C. D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology,* **48**, 443-453.

Nielsen, P. R., Ellgaard, L., Etzerodt, M., Thøgersen, H. C. and Poulsen, F. M. (1997) The solution structure of the N-terminal domain of α_2-macroglobulin receptor-associated protein. *Proceedings of National Academy of Science of the United States of America,* **94**, 7521-7525.

Noble, W. S. (2004) Support vector machine applications in computational biology, *Kernel Methods in Computational Biology,* MIT Press.

Notredame, C., Higgins, D. and Heringa, J. (2000) T-Coffee: A novel method for multiple sequence alignments. *Journal of Molecular Biology,* **302**, 205-217.

Olmea O, Valencia A (1997) Improving contact predictions by the combination of correlated mutations and other sources of sequence information. *Folding & Design* **2**, S25-S32.

Orengo, C. A., Brown, N. P. and Taylor, W. R. (1992). Fast structure alignment for protein databank searching. *Proteins: Structure, Function, and Genetics,* 14, 139-167.

Orengo, C. A., Sillitoe, I., Reeves, G. and Pearl, F. M. (2001) Review: what can structural classifications reveal about protein evolution? *Journal of Structural Biology*, **134**(2-3), 145-65.

Orengo, C. A., Michie, A. D., Jones, S., Jones, D. T., Swindells, M. B. and Thornton, J. M. (1997) CATH- a hierarchic classification of protein domain structures. *Structure* **5**, 1093-1108.

Osuna, E., Freund, R. and Girosi, F. (1997) An improved training algorithm for support vector machines. *Neural Networks for Signal processing VII-- proceeding of the 1997 IEEE workshop*, 276-285.

Park, B. and Levitt, M. (1995) The complexity and accuracy of discrete state models of protein structure. *Journal of Molecular Biology*, **249**(2), 493-507.

Park, B. and Levitt, M. (1996) Energy functions that discriminate X-ray and near-native folds from well-constructed decoys. *Journal of Molecular Biology*, **258**, 367-392.

Park, K. J. and Kanehisa, M. (2003) Prediction of protein subcellular locations by support verctor machines using compositions of amino acids and amino acid pairs. *Bioinformatics*, 19, 1656-1663.

Pavlidis, P., Weston, J., Cai, J. and Grundy, W. N. (2001) Gene functional classification from heterogeneous data. *Proceedings of the 5th International Conference on Computational Molecular Biology*, 242-248.

Peitsch, M. C. (1996) ProMod and Swiss-Model: Internet-based tools for automated comparative protein modeling. *Biochem Soc Trans*, **24**(1), 274-279.

Pollastri, G. and Baldi, P. (2002) Prediction of contact maps by GIOHMMs and recurrent neural networks using lateral propagation from all four cardinal corners. *Bioinformatics*, **18**, 62-70.

Qian, N. and Sejnowski, T. J. (1988) Predicting the secondary structure of globular proteins using neural network models. *Journal of Molecular Biology*, **202**, 865-884.

Ramachandran, G. N., Kolaskar, A. S., Ramakrishnan, C. and Sasisekharan, V. (1974) The mean geometry of the peptide unit from crystal structure data. *Biochim Biophys Acta*, **359**(2), 298-302.

Reczko, M. and Suhai, S. (1994) Applications of artificial neural networks in genome research. *Computational Methods in Genome Research*, ed. S. Suhai, 191-208, Plenum Press, New York.

Rice, D. W. and Eisenberg, D. (1997) A 3D-1D substitution matrix for protein fold recognition that includes predicted secondary structure of the sequence. *Journal of Molecular Biology*, **267**(4), 1026-1038.

Rost, B. and Sander, C. (1993) Prediction of protein secondary structure at better than 70% accuracy. *Journal of Molecular Biology*, **232**, 584-599.

Rost, B. Sander, C. and Schneider, R. (1994) PHD-an automatic mail server for protein secondary structure prediction. *Computer Applications in Biosciences*, **10**, 53-60.

Rost, B. (1995) TOPITS: threading one-dimensional predictions into three-dimensional structures. *Proc Int Conf Intell Syst Mol Biol*, **3**, 314-321.

Rost B, Casadio R, and Fariselli P. (1996) Refining neural network predictions for helical transmembrane proteins by dynamic programming. *Proc Int Conf Intell Syst Mol Biol.*, **4**, 192-200.

Rost, B., Schneider, R. and Sander, C. (1997) Protein fold recognition by prediction-based threading. *Journal of Molecular Biology*, **270**, 471-480.

Rost, B. (1999) Twilight zone of protein sequence alignments. *Protein Engineering*, **12**(2), 85-94.

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986) Learning internal representations by error propagation. Chapter 8 in *Parallel Distributed Processing: Foundation*. Vol. 1, MIT Press, Cambridge, MA, 318-362.

Russell, R. B., Copley, R. R. and Barton, G. J. (1996) Protein fold recognition by mapping predicted secondary structures. *Journal of Molecular Biology*, **259**(3), 349-65.

Samudrala, R. and Moult, J. (1998) An all-atom distance-dependent conditional probability discriminatory function for protein structure prediction. *Journal of Molecular Biology*, **275**, 895-916.

Samudrala, R., Huang, E. S., Levitt, M. (1998) Selection of the most native-like conformations from a set of models constructed by homology modelling. Unpublished results.

Samudrala, R., Xia, Y., Levitt, M., Huang, E. S. (1999) A combined approach for ab initio construction of low resolution protein tertiary structures from sequence. *Proceedings of the Pacific Symposium on Biocomputing*, 505-516.

Samudrala, R. and Levitt, M. (2000) Decoys 'R' Us: a database of incorrect conformations to improve protein structure prediction. *Protein Science*, 9, 1399-1401.

Samudrala, R. and Levitt, M. (2002) A comprehensive analysis of 40 blind protein structure predictions. *BMC Structural Biology*, 2, 3-18.

Schwede, T., Kopp, J., Guex, N. and Peitsch, M. C. (2003) SWISS-MODEL: an automated protein homology-modeling server. *Nucleic Acids Research*, 31, 3381-3385.

Segal, N. H., Pavlidis, P. Antonescu, C. R., Maki, R. G., Noble, W.S., Woodruff, J. M., Lewis, J. J., Brennan, M. F., Houghton, A. N. and Cordon-Cardo, C. (2003a) Classification and subtype prediction of soft tissue sarcoma by functional genomics and support vector machine analysis. *American Journal of Pathology*. 169:691-700.

Segal, N. H., Pavlidis, P., Noble, W.S., Antonescu, C. R., Viale, A., Wesley, U. V., Busam, K., Gallardo, H., DeSantis, D., Brennan, M. F., Cordon-Cardo, C. and Houghton, A. N. (2003b) Classification of clear cell sarcoma as melanoma of soft parts by genomic profiling. *Journal of Clinical Oncology*, 21:1775-1781.

Shan, Y. B., Wang, G. L. and Zhou, H. X. (2001) Fold recognition and accurate query-template alignment by a combination of PSI-BLAST and threading. *Proteins: structure, function, and bioinformatics*, **42**, 23-37.

Shi, J., Blundell, T. L. and Mizuguchi, K. (2001) FUGUE: sequence-structure homology recognition using environment-specific substitution tables and structure- dependent gap penalties. *Journal of Molecular Biology*, **310**, 243-257.

Shih, E. S. C. and Hwang, M. J. (2003) Protein structure comparison by probability-based matching of secondary structure elements. *Bioinformatics*, **19**, 735-741.

Simons, K. T., Kooperberg, C., Huang, E. and Baker, D. (1997) Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. *Journal of Molecular Biology*, **268**(1), 209-225.

Simons, K. T., Bonneau, R., Ruczinski, I. I. And Baker, D. (1999) Ab initio protein structure prediction of CASP □ targets using ROSETTA. *Proteins: structure, function, and bioinformatics*, **37**(S3), 171-176.

Sippl, M. J. (1990) Calculation of conformational ensembles from potentials of mean force. An approach to the knowledge-based prediction of local structures in globular proteins. *Journal of Molecular Biology*, **213**, 859-883.

Sippl, M. J. (1995) Knowledge-based potentials for proteins. *Current Opinion in Structural Biology*, **5**(2), 229-235.

Skolnick, J., Kolinski, A. and Ortiz, A. (2000) Derivation of protein-specific pair potentials based on weak sequence fragment similarity. *Proteins: structure, function, and bioinformatics*, **38**, 3-16.

Skolnick, J. and Kihara, D. (2001) Defrosting the frozen approximation: PROSPECTOR- a new approach to threading, *Proteins: structure, function, and bioinformatics*, 42: 319-331.

Skolnick, J. and Kolinski, A. (1991) Dynamic Monte Carlo simulations of a new lattice model of globular protein folding, structure and dynamics. *Journal of Molecular Biology*, **221**(2), 499-531.

Smith, T. F. and Waterman, M. S. (1981) Identification of common molecular subsequences. *Journal of Molecular Biology*, **147**, 195-197.

Snyder, E. E., Stormo, G. D. (1995) Identification of protein coding regions in genomic DNA. *Journal of Molecular Biology*, **248**, 1-18.

Sternberg, M. J. E. (1996) *Protein Structure Prediction: A Practical Approach*, Oxford University Press.

Stoesser, G., Bakwe, W., van den Broek, A., Camon, E., Garcia-Pastor, M., Kanz, C., Kulikova, T., Lombard, V., Lopez, R., Parkinson, H., Redaschi, N., Sterk, P., Stoehr, P. and Tuli, M. A. (2001) The EMBL nucleotide sequence database. *Nucleic acids research*, **10**, 2997-3011.

Stormo, G. D., Schneider, T. D., Gold, L. M. and Ehrenfeucht, A. (1982) Use of the "perceptron" algorithm to distinguish translational initiation sites in e. coli.. *Nucleic acids research*, **10**, 2997-3011.

Su, Y., Mural, M., Pavlovic, V. Schaffer, M. and Kasif, S. (2003) RankGene: Identification of diagnostic genes based on expression data. *Bioinformatics*, 19(12): 1578.

Sun, S. (1993) Reduced representation model of protein structure prediction: statistical potential and genetic algorithms, *Protein Science*, 2 (5), 762-785.

Taylor, W. R. and Orengo, C. A. (1989) Protein structure alignment. *Journal of Molecular Biology*, **208**(1), 1-22.

Taylor W.R. (1997) Multiple sequence threading: an analysis of alignment quality and stability. *Journal of Molecular Biology*, **269**, 902-943.

Thiele, R., Zimmer, R. and Lengauer, T. (1999) Protein threading by recursive dynamic programming. *Journal of Molecular Biology*, **290**, 757-779.

Thompson, J. D., Higgins, D. G. and Gibson, T. J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, **22**:4673-4680.

Thornton, J. M., Orengo, C. A., Todd, A. E. and Pearl, F. M. (1999) Protein folds, functions and evolution. *Journal of Molecular Biology*, **293**(2), 333-42.

Torda, A. E., Procter, J. B. and Huber, T. (2004) Wurst: a protein threading server with a structural scoring function, sequence profiles and optimized substitution matrices, *Nucleic acids research*, **32**, W532-W535.

Uberbacher, E. C. and Mural, R. J. (1991) Locating protein-coding regions in human DNA sequences by a multiple sensor-neural network approach. *Proceedings of the National Academy of Science of the U. S. A.*, **88**, 11261-11265.

Unger, R. and Moult, J. (1991) An analysis of protein folding pathways. *Biochemistry*, **30**, 3816-3823.

Vapnik, V. (1995) *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.

Vapnik, V. N. (1998) Statistical Learning Theory, Adaptive and learning systems for signal processing, communications, and control. Wiley, New York.

Vendruscolo, M., Najmanovich, R. and Domany, E. (2000) Can a pairwise contact potential stabilize native protein folds against decoys obtained by threading? *Proteins: structure, function, and bioinformatics*, **38**, 134-148.

Vert, J. P. and kanehisa, M. (2003) Graph-driven features extraction from microarray data using diffusion kernels and kernel CCA. *Advances in Neural Information Processing Systems 15*. MIT Press, Cambridge, MA.

Wallner, B., Fang, H., Ohlson, T., Frey-Skott, J. and Elofsson, A. (2004) Using evolutionary information for query and target improves fold recognition. *Proteins: structure, function, and bioinformatics*, **54**:342-350.

Wilcox, G. L., Poliac, M. O. and Liebman, M. N. (1991) Neural network analysis of protein tertiary structure. *Tetrahedron Computer Methods*, **3**, 191-211.

Williams, M.G., Shirai, H., Shi, J. et al. (2001) Sequence-structure homology recognition by iterative alignment refinement and comparative modeling. *Proteins: structure, function, and bioinformatics*, **S5**, 92-97.

Wu, C. H. (1997) Artificial neural networks for molecular sequence analysis. *Computers & Chemistry*, **21**(4), 237 - 256.

Xia, Y., Huang, E. S., Levitt, M., Samudrala, R. (2000) Ab initio construction of protein tertiary structures using a hierarchical approach. *Journal of Molecular Biology*, 300, 171-185.

Xin, Y., Carmeli, T. T., Liebman, M. N. and Wilcox, G. L. (1993) Use of the backpropagation neural network algorithm for prediction of protein folding patterns. In *Proceedings of the Second International Conference on Bioinformatics, Supercomputing, and Complex Genome analysis*, eds H. A. Lim, J. W. Fickett, C. R. Cantor and R. J. Robbins, 359-375. World Scientific, River Edge, NJ.

Xu, D., Crawford, O. H., LoCascio, p. F. and Xu Y. (2001) Application of PROSPECT in CASP4: Characterizing protein structures with new folds. *Proteins: structure, function, and bioinformatics*, **S5**, 140.

Xu, Y. and Xu, D. (2000) Protein threading using PROSPECT: design and evaluation. *Proteins: structure, function, and bioinformatics*, **40**(3), 343-354.

Xu, Y., Xu, D., and Olman, V. (2002) A Preactical Method for Interpretation of Threading Scores: An Application of neural Network. *Statistica Sinica*, **12**, 159-177.

Yeang, C., Ramaswamy, S., Tamayo, P., Mukherjee, S., Rifkin, R. R., Angelo, M., Reich, M., Lander, E., Mesirov, J. and Golub, T. (2001) Molecular classification of multiple tumor types. *Bioinformatics*, 17, Suppl 1:S316-S322.

Yona, G and Levitt, M. (2002) Within the twilight zone: a sensitive profile-profile comparison tool based on information theory. *Journal of Molecular Biology*, **315**, 1257-1275.

Zavaljevski, N. and Reifman, J. (2002) Support vector machines with selective kernel scaling for protein classification and identification of key amino acid positions. *Bioinformatics*, 18(5):698-696.

Zhang, K. Y. and Eisenberg, D. (1994) The three-dimensional profile method using residue preference as a continuous function of residue environment. *Protein Science*, **3**(4), 687-695.

Zhou, H and Zhou, Y. (2002) Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction, *Protein Science*, 11, 2714-2726.

Zhou, H. and Zhou, Y. (2004) Single-body residue-level knowledge-based energy score combined with sequence-profile and secondary structure information for fold recognition. *Proteins: structure, function, and bioinformatics*, **55**:1005-1013.

Zien, A., Ratch, G., Mika, S., Scholkopf, B., Legauer, T. and Muller, K. R. (2000) Engineering support vector machine kernels that recognize translation initiation sites. *Bioinformatics*, 16, 799-807.

APPENDIX I SIDE CHAIN RADIUS OF AMINO ACIDS

The calculation of side chain radius is as follows:

For all the amino acids, the residue messes are known. The main chain mass for all is 56.0D. So, the residue side chain mess equals residue mess minus main chain mess. For example, the residue mess of an Alanine is 71.1D. The side chain mess is 71.1-56=15.1D.

The radius of an Alanine side-chain sphere is known as 1.7 $\mathring{A}$, and the radius is supposed to be proportional to the cube roots of its mass, the proportional value is $1.45 = \sqrt[3]{15.1}/1.7$.

Thus, the side chain radius of other amino acids could be computed and list in Table 1. For example, Histidine (HIS), the side chain mess is 137.1-56=81.1D, the side chain radius=$\sqrt[3]{81.1}/1.45$=2.98.

Amino acid	Residue mass(D)	Side-chain mass(D)	Side-chain radius(A)
GLY(G)	57	1.0	0.69
ALA(A)	71.1	15.1	1.70
VAL(V)	99.1	43.1	2.41
LEU(L)	113.2	57.2	2.65

ILE(I)	113.2	57.2	2.65
MET(M)	131.2	75.2	2.90
PRO(P)	97.1	41.1	2.37
PHE(F)	147.2	91.2	3.10
TRP(W)	186.2	130.2	3.49
SER(S)	87.1	31.1	2.16
THR(T)	101.1	45.1	2.45
ASN(N)	114.1	58.1	2.66
GLN(Q)	128.1	72.1	2.86
TYR(Y)	163.2	107.2	3.27
CYS(C)	103.1	47.1	2.48
LYS(K)	128.2	72.2	2.86
ARG(R)	156.2	100.2	3.19
HIS(H)	137.1	81.1	2.98
ASP(D)	115.1	59.1	2.68
GLU(E)	129.1	73.1	2.88

Table 1 The Side-chain radius of amino acid residue

APPENDIX Π AN EXAMPLE OF PROTEIN STRUCTURAL ALIGNMENT BY FLASH

FLASH (v1.0) is a package for finding similarity in the three-dimensional structures of proteins. It can be used to compare one protein structure against another. So, for each protein data pair we selected from SCOP, FLASH can be implemented to give a result of structural alignment.

Aligning protein pairs with FLASH, two kinds of files are required. They are 'pdb' file and 'sse' file. Here we take a protein pair, 1bfd_1 and 1d4oa, as an example to show how to align the two proteins with FLASH.

First, 'pdb' files are built for each protein. FLASH requires the only CA atom records form the protein data stored in PDB, thus each 'pdb' file with the information of CA atoms are extracted from the files in PDB. The 1bfd_1.pdb is shown in following.

1bfd_1.pdb

```
ATOM   1384  CA  SER   182      78.554  17.521 143.677  1.00 14.63
ATOM   1392  CA  VAL   183      80.804  16.472 140.831  1.00 14.27
ATOM   1399  CA  ARG   184      84.105  14.696 140.396  1.00 14.67
ATOM   1410  CA  LEU   185      86.687  13.990 137.700  1.00 16.11
ATOM   1418  CA  ASN   186      85.559  11.595 134.938  1.00 16.59
ATOM   1426  CA  ASP   187      86.586   7.922 135.256  1.00 19.65
ATOM   1434  CA  GLN   188      89.359   7.895 132.662  1.00 25.49
ATOM   1443  CA  ASP   189      91.237  10.899 134.046  1.00 21.66
ATOM   1451  CA  LEU   190      90.548   9.859 137.626  1.00 21.79
ATOM   1459  CA  ASP   191      92.210   6.512 136.914  1.00 26.16
ATOM   1467  CA  ILE   192      95.272   8.349 135.597  1.00 21.93
```

```
ATOM   1475  CA  LEU   193      95.483  10.481 138.756  1.00 20.29
ATOM   1483  CA  VAL   194      95.092   7.385 140.981  1.00 21.14
.....................
.....................
ATOM   2440  CA  LEU   325      90.390   9.131 145.962  1.00 20.36
ATOM   2448  CA  ALA   326      90.265   7.574 149.456  1.00 22.80
ATOM   2453  CA  ASN   327      88.413   4.562 148.060  1.00 26.32
ATOM   2461  CA  LEU   328      90.443   4.018 144.890  1.00 23.12
ATOM   2469  CA  VAL   329      94.090   4.311 145.974  1.00 23.00
ATOM   2476  CA  GLU   330      95.998   1.152 146.894  1.00 27.78
ATOM   2485  CA  GLU   331      97.393   0.687 150.383  1.00 30.72
ATOM   2494  CA  SER   332     101.096   1.490 150.204  1.00 28.60
ATOM   2500  CA  SER   333     103.587  -0.957 151.710  1.00 30.04
ATOM   2506  CA  ARG   334     105.526   2.031 153.079  1.00 26.45
ATOM   2517  CA  GLN   335     105.622   2.477 156.841  1.00 23.31
ATOM   2531  CA  LEU   336     103.081   4.899 158.318  1.00 25.42
ATOM   2539  CA  PRO   337     104.717   8.206 159.370  1.00 24.31
ATOM   2546  CA  THR   338     105.601   8.560 163.060  1.00 25.03
ATOM   2556  CA  ALA   339     103.621  11.092 165.085  1.00 24.61
ATOM   2561  CA  ALA   340     105.279  14.389 165.918  1.00 25.05
ATOM   2566  CA  PRO   341     106.569  14.552 169.510  1.00 28.13
```

Second, the 'sse' files are created for each protein. For each 'sse' file, the secondary structure information of Helix and Strand are extracted from the files in PDB. The example of sse files for 1bfd_1 and 1d4oa are shown:

1bfd_1.sse

```
H 187 199 209 213 217 227 257 264 302 307 318 328
S 204 207 230 233 249 252 269 273 294 299 312 315
```

1d4oa.sse

```
H 11 22 30 36 38 52 69 79 81 83 88 92 93 98 107 111 112 117 129 133
152 157
S 24 29 55 60 85 87 100 104 136 140 160 164
```

Taking 1bfd_1 as an example, the 'sse' file means that the protein (1bfd_1) contains six helices (187-199, 209-213, 217-227, 257-264, 302-307 and 318-328) and six strands (204-207, 230-233, 249-252, 269-273, 294-299 and 312-315).

Third, after creating the 'pdb' and 'sse' files for the two proteins, FLASH program is executed to give the output of the structural alignment. The example of the alignment for 1bdfd_1 and 1d4oa is given below:

```
First:1bfd_1.pdb Residue: 160 Helix: 6 Strand: 6
Second:1d4oa_.pdb Residue: 177 Helix:11 Strand: 6
Total_solutions: 1
```

```
No. #AlnSSE Rough_Z Refine_Z p-value #AlnRes    RMSD  Seq%Id
  1       9    6.16    24.78 1.7e-11     123    2.05      21



No.  1 residue_alignment:
        ID  1234567890 1234567890 1234567890 1234567890 1234567890

                  hhhhh hhhhhhhh     bbbb hhh hh    hhhhh hhhhhh
1bfd_1.pdb: SVRLNDQDLD ILVKALNSAS NPAIVLGPDV DAANANADCV MLAERLK---
                    :|| :::::::::: :::|::|::: :||:|::::: :|:::|:
1d4oa_.pdb: -GTHTEINLD NAIDMIREAN SIIITPGYGL CAAKAQYPIA DLVKMLSEQG
                  hhh hhhhhhhhhh  bbbbbbhhhh hhh hhhhhh hhhhhhhhh


                bbbb                              bbbb      hhhhhhhh
1bfd_1.pdb: --APVWVAP- SAP----RC- -PFP--TRH- PCFRGLMPAG IAAISQLLEG
              ::::: :  :|:    ::   :::    :  :::: :|:       : :::::
1d4oa_.pdb: KKVRFGI-HP VAGRMPGQLN VLLAEAGVPY DIVL-EMD-- -EIN-HDFPD
             bbbbbb          hhhh hhhhhhh      bb bhh     hhh hhhhh


              bbbbb                                        bbbbbb  h
1bfd_1.pdb: HDVVLVIGAP --VF-RY--- -------HQY DPGQY-LKPG TRLISVTCDP
            :|:|||||||:   |:  :           ::      : :| : :::|::::::
1d4oa_.pdb: TDLVLVIGAN DTVNSAAQED PNSIIAGMP- --VLEVWK-S KQVIVMKRSL
            h bbbbb  h hhhhhhhhhh                 hhhh h   bbbbb


            hh   hhh          bbb b  hhhhhhh hhhh
1bfd_1.pdb: LE---AARAP M-----GDAI VADIGAMASA LANLV--EES SRQLPTAAP
            ::     ::| :     :::: ::|:::::::| |:::|   :
1d4oa_.pdb: GVGYAAVDNP IFYKPNTAML LGDAKKTCDA LQAKVRES-- ---------
                  hh hhhh  bbbb b hhhhhhhh hhhhhhh
```

APPENDIX III THE NEURAL NETWORK TRAINING RESULTS FOR TES MODEL

The BPNN for building TES model is trained by using various network architectures with the number of neurons in hidden layer and different starting conditions. The average training and test error is shown in Table 2. The best performance architecture of NN is the one with 22 hidden neurons. It can be seen from Table 2 that the one with 22 hidden neurons' network has the minimum average test error of 2.648. Table 3 showed the training and test error of the NNs (22 hidden neurons) with 10 times different initializations. The 2.NN is adopted as trained NN for benchmark problem evaluation due to its best performance.

Number	Hidden neuron	Average training error	Average test error
1	10	2.64333	2.67409
2	12	2.63922	2.67396
3	14	2.62806	2.66284
4	16	2.64108	2.67862
5	18	2.62466	2.66170
6	20	2.62209	2.65798
7	**22**	**2.61111**	**2.64841**
8	24	2.61474	2.64922
9	26	2.62078	2.65889
10	28	2.64066	2.66782
11	30	2.64704	2.67051

Table 2 The training and test error for different ANN architectures

NN name	Training error	Test error
0.NN	2.62580	2.65477
1.NN	2.62134	2.65658
2.NN	**2.61602**	**2.65162**
3.NN	2.62540	2.66203
4.NN	2.62750	2.66080
5.NN	2.62644	2.66113
6.NN	2.63178	2.67051
7.NN	2.62813	2.66424
8.NN	2.61284	2.65005
9.NN	2.61589	2.65943

Table 3 The training and test error for different initialise

In Table 3, the training and testing relative entropy errors of the 2.NN model are 2.616 and 2.652. Figure 1 shows the curve of training error. The training is

stopped at 429 epochs by using ten-fold cross validation approach and the error is 2.616.

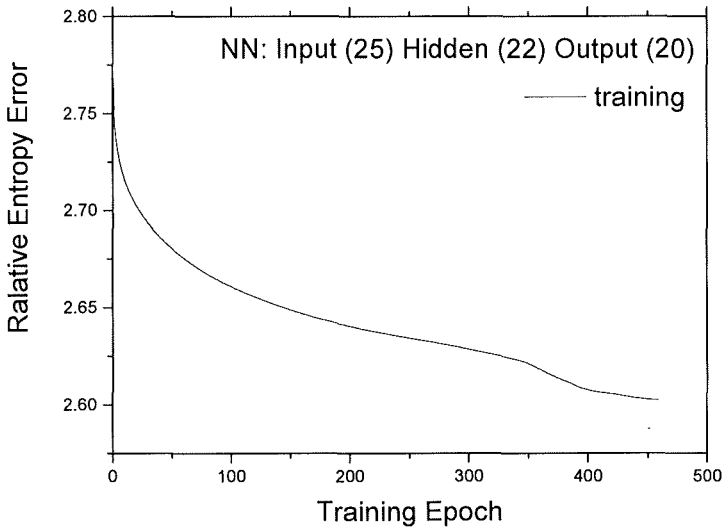


Figure 1 Relative entropy errors of training. The training stopped at 429 epochs and the error is 2.62.

APPENDIX IV THE TESTING RESULTS OF TES ON PROSTAR DECOY SETS

There are three decoy sets in PROSTAR website. They are misfold, asilomar and ifu. The performance of our TES model is tested on discrimination the native and decoys. The testing results are shown in the following. For each pairs of native/decoy proteins, the compatibility scores are computed by our TES model. If the compatibility score of native is larger than decoy, then it means the TES model can correctly distinguish the native from decoy. The result is correct. If the compatibility score of native is smaller than decoy, then the TES model is not able to discriminate native and decoy. The result is wrong.

Asilomar

Native	Compatibility	decoy	Compatibility	Result
CRYSTAL.CRABPI.PDB[a]	43.7982[b]	CRABPI-ABAGYAN.PDB[c]	22.8044[d]	Correct
CRYSTAL.CRABPI.PDB	43.7982	CRABPI-MOULT1.PDB	31.9539	Correct
CRYSTAL.CRABPI.PDB	43.7982	CRABPI-MOULT2.PDB	30.6625	Correct
CRYSTAL.CRABPI.PDB	43.7982	CRABPI-SALI.PDB	34.5877	Correct
CRYSTAL.CRABPI.PDB	43.7982	CRABPI-VINALS1.PDB	30.1783	Correct
CRYSTAL.CRABPI.PDB	43.7982	CRABPI-VINALS2.PDB	41.3005	Correct
CRYSTAL.CRABPI.PDB	43.7982	CRABPI-VINALS3.PDB	33.6194	Correct
CRYSTAL.CRABPI.PDB	43.7982	CRABPI-VRIEND.PDB	23.2239	Correct
CRYSTAL.CRABPI.PDB	43.7982	CRABPI-WEBER1.PDB	38.3004	Correct

CRYSTAL.CRABPI.PDB	43.7982	CRABPI-WEBER2.PDB	35.9954	Correct
CRYSTAL.EDN.PDB	31.5908	EDN-BIOSYM.PDB	16.7033	Correct
CRYSTAL.EDN.PDB	31.5908	EDN-KOEHL.PDB	22.939	Correct
CRYSTAL.EDN.PDB	31.5908	EDN-MOULT.PDB	22.0546	Correct
CRYSTAL.EDN.PDB	31.5908	EDN-SALI1.PDB	23.0551	Correct
CRYSTAL.EDN.PDB	31.5908	EDN-SALI2.PDB	22.6822	Correct
CRYSTAL.EDN.PDB	31.5908	EDN-SAQI1.PDB	13.6662	Correct
CRYSTAL.EDN.PDB	31.5908	EDN-SAQI2.PDB	16.6963	Correct
CRYSTAL.EDN.PDB	31.5908	EDN-VINALS1.PDB	11.8171	Correct
CRYSTAL.EDN.PDB	31.5908	EDN-VINALS2.PDB	16.2098	Correct
CRYSTAL.EDN.PDB	31.5908	EDN-VINALS3.PDB	23.6921	Correct
CRYSTAL.EDN.PDB	31.5908	EDN-WEBER.PDB	11.5041	Correct
CRYSTAL.HALOFER.PDB	35.5703	HALOFER-WEBER.PDB	14.3073	Correct
CRYSTAL.MCHPR.PDB	29.7597	MCHPR-ABAGYAN.PDB	29.9659	Wrong
CRYSTAL.MCHPR.PDB	29.7597	MCHPR-BIOSYM.PDB	25.6795	Correct
CRYSTAL.MCHPR.PDB	29.7597	MCHPR-KOBAYASHI.PDB	3.62302	Correct
CRYSTAL.MCHPR.PDB	29.7597	MCHPR-KOEHL1.PDB	25.4808	Correct
CRYSTAL.MCHPR.PDB	29.7597	MCHPR-KOEHL2.PDB	26.2762	Correct
CRYSTAL.MCHPR.PDB	29.7597	MCHPR-MOSENKIS.PDB	30.2057	Wrong
CRYSTAL.MCHPR.PDB	29.7597	MCHPR-MOULT.PDB	27.7987	Correct
CRYSTAL.MCHPR.PDB	29.7597	MCHPR-VIHINEN.PDB	25.7908	Correct
CRYSTAL.MCHPR.PDB	29.7597	MCHPR-VRIEND.PDB	28.4586	Correct
CRYSTAL.MCHPR.PDB	29.7597	MCHPR-WEBER.PDB	31.3352	Wrong
CRYSTAL.NDK.PDB	43.7147	NDK-ABAGYAN.PDB	43.6039	Correct
CRYSTAL.NDK.PDB	43.7147	NDK-KOEHL.PDB	47.4615	Wrong
CRYSTAL.NDK.PDB	43.7147	NDK-SALI.PDB	40.4217	Correct
CRYSTAL.NDK.PDB	43.7147	NDK-VIHENEN.PDB	50.3416	Wrong
CRYSTAL.NDK.PDB	43.7147	NDK-VRIEND.PDB	45.6966	Wrong
CRYSTAL.NDK.PDB	43.7147	NDK-WEBER1.PDB	42.1895	Correct
CRYSTAL.NDK.PDB	43.7147	NDK-WEBER2.PDB	43.8578	Wrong
CRYSTAL.P450.PDB	143.114	P450-ABAGYAN.PDB	95.0603	Correct
CRYSTAL.P450.PDB	143.114	P450-WEBER.PDB	98.8035	Correct

Table 4 The Compatibility score of native and decoy pairs in Asilomar set

[a] The name of native protein.

[b] The compatibility value of native protein, which is calculated by adding all the compatibility scores of every residue in protein sequence.

[c] The name of decoy protein.

[d] The compatibility value of decoy protein. If the value in this column is less than that of native protein ([b] column), then it means the model could successfully distinguish the native and decoy protein and vice versa.

Misfold

Native	Compatibility	Decoy	Compatibility	Result
CRYSTAL.1BP2.PDB	27.1196	1BP2ON2PAZ.PDB	-9.62587	Correct
CRYSTAL.1CBH.PDB	20.3959	1CBHON1PPT.PDB	-1.09364	Correct
CRYSTAL.1FDX.PDB	19.0866	1FDXON5RXN.PDB	-6.816	Correct
CRYSTAL.1HIP.PDB	30.6937	1HIPON2B5C.PDB	-10.5216	Correct
CRYSTAL.1LH1.PDB	48.7715	1LH1ON2I1B.PDB	1.0774	Correct
CRYSTAL.1P2P.PDB	30.0617	1P2PON1RN3.PDB	-1.04321	Correct
CRYSTAL.1PPT.PDB	-3.64951	1PPTON1CBH.PDB	-11.7326	Correct
CRYSTAL.1REI.PDB	34.0123	1REION5PAD.PDB	-9.98275	Correct
CRYSTAL.1RHD.PDB	104.336	1RHDON2CYP.PDB	26.5752	Correct
CRYSTAL.1RN3.PDB	31.6757	1RN3ON1P2P.PDB	-0.585725	Correct
CRYSTAL.1SN3.PDB	32.805	1SN3ON2CI2.PDB	-0.837851	Correct
CRYSTAL.1SN3.PDB	32.805	1SN3ON2CRO.PDB	-6.53431	Correct
CRYSTAL.2B5C.PDB	25.6586	2B5CON1HIP.PDB	-3.23917	Correct
CRYSTAL.2CDV.PDB	21.0004	2CDVON2SSI.PDB	0.208361	Correct
CRYSTAL.2CI2.PDB	9.4209	2CI2ON1SN3.PDB	-9.19357	Correct
CRYSTAL.2CI2.PDB	9.4209	2CI2ON2CRO.PDB	-11.8764	Correct
CRYSTAL.2CRO.PDB	13.3881	2CROON1SN3.PDB	-12.8286	Correct
CRYSTAL.2CRO.PDB	13.3881	2CROON2CI2.PDB	-10.7121	Correct
CRYSTAL.2CYP.PDB	101.693	2CYPON1RHD.PDB	26.4458	Correct
CRYSTAL.2I1B.PDB	41.69	2I1BON1LH1.PDB	6.82112	Correct
CRYSTAL.2PAZ.PDB	46.2042	2PAZON1BP2.PDB	5.72203	Correct
CRYSTAL.2SSI.PDB	30.3193	2SSION2CDV.PDB	-4.3581	Correct
CRYSTAL.2TMN.PDB	95.1398	2TMNON2TS1.PDB	15.9299	Correct
CRYSTAL.2TS1.PDB	113.356	2TS1ON2TMN.PDB	-9.30169	Correct

CRYSTAL.5PAD.PDB	67.5755	5PADON1REI.PDB	-24.0963	Correct

Table 5 The Compatibility score of native and decoy pairs in Misfold set

ifu

Native	Compatibility	Decoy	Compatibility	Result
CRYSTAL.1ALC_21-32.PDB	-8.54096	1ALC_21-32.PDB	-8.60252	Correct
CRYSTAL.1ALC_21-36.PDB	-7.06592	1ALC_21-36.PDB	-8.61842	Correct
CRYSTAL.1BGS_10-22.PDB	-2.51359	1BGS_10-22.PDB	-4.98938	Correct
CRYSTAL.1BGS_88-98.PDB	-8.49428	1BGS_88-98.PDB	-6.45713	Wrong
CRYSTAL.1FKF_27-38.PDB	3.37854	1FKF_27-38.PDB	2.92278	Correct
CRYSTAL.1FKF_46-59.PDB	-5.50734	1FKF_46-59.PDB	-6.4245	Correct
CRYSTAL.1FKF_46-61.PDB	-3.64236	1FKF_46-61.PDB	-4.47646	Correct
CRYSTAL.1HGF_100-113.PDB	-3.94131	1HGF_100-113.PDB	-4.34226	Correct
CRYSTAL.1HRC_7-18.PDB	-5.53031	1HRC_7-18.PDB	-8.26554	Correct
CRYSTAL.1HRC_92-103.PDB	-3.25679	1HRC_92-103.PDB	-2.95445	Wrong
CRYSTAL.1ILB_99-110.PDB	-5.22427	1ILB_99-110.PDB	-4.92366	Wrong
CRYSTAL.1LMB_15-26.PDB	0.257263	1LMB_15-26.PDB	0.110398	Correct
CRYSTAL.1MBC_131-142.PDB	-3.19509	1MBC_131-142.PDB	-3.83852	Correct
CRYSTAL.1MBC_131-146.PDB	-3.10427	1MBC_131-146.PDB	-3.54935	Correct
CRYSTAL.1MBC_29-40.PDB	-4.95975	1MBC_29-40.PDB	-5.12092	Correct
CRYSTAL.1MBC_29-43.PDB	-5.57476	1MBC_29-43.PDB	-12.0616	Correct
CRYSTAL.1MBC_6-17.PDB	-7.04666	1MBC_6-17.PDB	-7.2428	Correct
CRYSTAL.1MBC_6-21.PDB	-6.35491	1MBC_6-21.PDB	-6.46396	Correct
CRYSTAL.1MBC_99-111.PDB	-7.08498	1MBC_99-111.PDB	-10.2439	Correct
CRYSTAL.1MBC_99-119.PDB	-10.9675	1MBC_99-119.PDB	-13.3796	Correct
CRYSTAL.1PGA_43-54.PDB	-5.47716	1PGA_43-54.PDB	-4.53947	Wrong
CRYSTAL.1UBQ_1-17.PDB	-6.03729	1UBQ_1-17.PDB	-4.74754	Wrong
CRYSTAL.1UBQ_26-41.PDB	5.39729	1UBQ_26-41.PDB	2.19549	Correct
CRYSTAL.1UBQ_3-15.PDB	-5.08568	1UBQ_3-15.PDB	-6.17751	Correct
CRYSTAL.2I1B_100-115.PDB	-4.25778	2I1B_100-115.PDB	-6.06304	Correct
CRYSTAL.2I1B_103-112.PDB	-3.73616	2I1B_103-112.PDB	-4.48068	Correct
CRYSTAL.2I1B_69-82.PDB	-7.39211	2I1B_69-82.PDB	-3.55183	Wrong
CRYSTAL.2MHR_102-113.PDB	-5.91935	2MHR_102-113.PDB	-6.3745	Correct

CRYSTAL.2MHR_44-59.PDB	-4.67574	2MHR_44-59.PDB	-4.67652	Correct
CRYSTAL.2MHR_51-62.PDB	-3.18436	2MHR_51-62.PDB	-2.71314	Wrong
CRYSTAL.2MHR_52-67.PDB	-1.73013	2MHR_52-67.PDB	-3.13186	Correct
CRYSTAL.2MHR_65-84.PDB	-3.70062	2MHR_65-84.PDB	-5.10793	Correct
CRYSTAL.2MHR_67-78.PDB	-2.2953	2MHR_67-78.PDB	-3.18056	Correct
CRYSTAL.2MHR_67-82.PDB	-4.21179	2MHR_67-82.PDB	-6.33454	Correct
CRYSTAL.2PCY_18-29.PDB	-5.23243	2PCY_18-29.PDB	-4.59381	Wrong
CRYSTAL.2PCY_41-56.PDB	5.48788	2PCY_41-56.PDB	-0.279805	Correct
CRYSTAL.3LZM_24-35.PDB	-4.66696	3LZM_24-35.PDB	-4.91039	Correct
CRYSTAL.3LZM_99-111.PDB	-3.10226	3LZM_99-111.PDB	-5.35682	Correct
CRYSTAL.3LZM_99-114.PDB	-2.33292	3LZM_99-114.PDB	-5.09899	Correct
CRYSTAL.3SNS_16-29.PDB	-4.78637	3SNS_16-29.PDB	-2.0131	Wrong
CRYSTAL.3SNS_6-21.PDB	-5.59337	3SNS_6-21.PDB	-5.64975	Correct
CRYSTAL.4PTI_22-33.PDB	-5.75293	4PTI_22-33.PDB	-4.16059	Wrong
CRYSTAL.5CYT_88-101.PDB	-0.235017	5CYT_88-101.PDB	-0.439144	Correct
CRYSTAL.7RSA_2-13.PDB	0.622406	7RSA_2-13.PDB	-0.891238	Correct

Table 6 The Compatibility score of native and decoy pairs in ifu set

APPENDIX V PUBLICATION LIST

- Nan Jiang, Wendy Xinyu Wu, Ian Mitchell (2005) Protein Fold Recognition Using Neural Networks and Support Vector Machines, Lecture Notes in Computer Science, Vol. 3578 (IDEAL 2005), 462-469.

- Nan Jiang, Wendy Xinyu Wu, Ian Mitchell (2005) Protein Fold Recognition by Mixed Environment-Specific Amino Acid Substitution Mapping Using Neural Networks, The first IEEE International Workshop on High Performance Computing in Medicine and Biology (HiPCoMB-2005), Proceeding of the Eleventh ICPADS, 341-345.

- Nan Jiang, Wendy Xinyu Wu and Ian Mitchell (2005) Threading with Environment-specific Score by Artificial Neural Networks, Soft Computing, In Press.

- Nan Jiang, Wendy Xinyu Wu and Ian Mitchell (2004) Protein Threading with Residue-environment Matching by Artificial Neural Networks , Proceedings of the 2004 ACM symposium on applied computing, Vol. 1, 209.